

The L^AT_EX 2_ε Sources

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

2020-02-01 Patch level -3

This file is maintained by the L^AT_EX Project team.
Bug reports can be opened (category `latex`) at
<https://latex-project.org/bugs.html>.

Contents

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

6	Filename Parsing	11
7	T_EX Versions	13
8	ltxcheck.tex	13
b	ltpplain.dtx	15
9	Plain T_EX	15
c	ltvers.dtx	33
10	Version Identification	33
d	ltdefns.dtx	36
11	Definitions	36
11.1	Initex initialisations	36
11.2	Saved versions of T _E X primitives	36
11.3	Command definitions	37
11.4	Robust commands and protect	44
11.5	Internal defining commands	48
12	Discretionary Hyphenation	51
e	ltdalloc.dtx	54
13	Counters	54
f	ltdcntrl.dtx	56
14	Program control structure	56
g	ltderror.dtx	60
15	Error handling and tracing	60
15.1	General commands	60
15.2	Specific errors	65
15.3	Tracing	68
h	ltdpar.dtx	69
16	Paragraphs	69
16.1	Implementation	69

i	ltspace.dtx	71
17	Spacing	71
17.1	User Commands	71
17.2	Chris' comments	71
17.3	Some immediate actions	73
17.4	The code	74
17.5	Vertical spacing	80
17.6	Horizontal space (and breaks)	84
j	ltlogos.dtx	87
18	Logos	87
k	ltfiles.dtx	88
19	File Handling	88
19.1	Safe Input Macros	96
19.2	Listing files	100
l	ltoutenc.dtx	102
20	Font encodings	102
20.1	Removing encoding-specific commands	104
20.2	The order of declarations	105
20.3	Docstrip modules	105
20.4	Definitions for the kernel	106
20.4.1	Declaration commands	106
20.4.2	Hyphenation	113
20.4.3	Miscellania	114
20.4.4	Default encodings	114
20.4.5	Math material	116
20.5	Definitions for the OT1 encoding	118
20.6	Definitions for the T1 encoding	120
20.7	Definitions for the OMS encoding	125
20.8	Definitions for the OML encoding	126
20.9	Definitions for the OT4 encoding	126
20.10	Definitions for the TS1 encoding	128
20.11	Definitions for the TU encoding	132
21	Package files	142
21.1	The fontenc package	142
m	ltcounts.dtx	145
22	Counters and Lengths	145
22.1	Environment Counter Macros	145

n	ltnlength.dtx	152
23	Lengths	152
p	ltfssbas.dtx	176
28	Preliminary macros	176
29	Macros for setting up the tables	177
30	Selecting a new font	182
30.1	Macros for the user	182
30.2	Macros for loading fonts	186
31	Assigning math fonts to <i>versions</i>	193
p	ltfssbas.dtx	176
28	Preliminary macros	176
29	Macros for setting up the tables	177
30	Selecting a new font	182
30.1	Macros for the user	182
30.2	Macros for loading fonts	186
31	Assigning math fonts to <i>versions</i>	193
q	ltfsstrc.dtx	199
32	Introduction	199
33	A driver for this document	199
34	The Implementation	200
35	Handling Options	200
36	Macros common to fam.tex and tracefnt.sty	202
36.1	General font loading	202
36.2	Math fonts setup	206
36.2.1	Outline of algorithm for math font sizes	206
36.2.2	Code for math font size setting	208
36.2.3	Other code for math	209
37	Scaled font extraction	211
37.1	Sizefunctions	218

r	ltfsscmp.dtx	222
s	ltfssdcl.dtx	226
38	Interface Commands	226
t	ltfssini.dtx	252
39	NFSS Initialisation	252
39.1	Providing math <i>versions</i>	252
40	Custom series settings for main document families	253
41	Supporting nested emphasis	261
41.1	Miscellaneous	264
u	fontdef.dtx	268
42	Introduction	268
43	Customization	268
44	The docstrip modules	269
45	A driver for this document	269
46	The fonttext.ltx file	270
46.1	Encodings	270
46.2	Defaults	272
47	The fontmath.ltx file	274
47.1	The font encodings used	274
47.1.1	Symbolfont and Alphabet declarations	274
47.2	Math font sizes	275
47.3	The math symbol assignments	275
47.3.1	The letters	275
47.3.2	The digits	276
47.3.3	Punctuation, brace, etc. keys	277
47.3.4	Delimitercodes for characters	277
47.4	Symbols accessed via control sequences	278
47.4.1	Greek letters	278
47.4.2	Ordinary symbols	279
47.4.3	Large Operators	279
47.4.4	Binary symbols	280
47.4.5	Relations	280
47.4.6	Arrows	282
47.4.7	Punctuation symbols	283
47.4.8	Math accents	283
47.4.9	Radicals	284

47.4.10 Over and under something, etc	284
47.4.11 Delimiters	284
47.5 Math versions of text commands	286
47.6 Other special functions and parameters	286
47.6.1 Biggggg	286
47.6.2 The log-like functions	286
47.6.3 Parameters	287
48 Default cfg files	287
v preload.dtx	288
49 Overview	288
50 Customization	288
51 Module switches for the DOCSTRIP program	289
52 A driver for this document	289
53 The code	289
w ltfntcmd.dtx	292
54 Introduction	292
55 The implementation	294
56 Initialization	300
x ltttextcomp.dtx	301
57 Sub-encodings	305
57.1 Sub-encoding 1 (drop symbols not working in Latin Modern) . . .	306
57.2 Sub-encoding 2 (majority of new OTF fonts via autoinst) . . .	306
57.3 Sub-encoding 3	309
57.4 Sub-encoding 4	309
57.5 Sub-encoding 5 (most older PS fonts)	309
57.6 Sub-encoding 6	309
57.7 Sub-encoding 7	309
57.8 Sub-encoding 8	310
57.9 Sub-encoding 9 (most missing)	310
58 Unicode engine specials	310
59 Font family sub-encodings setup	311
60 Legacy symbol support for lists and footnote symbols	315

61 The textcomp package	319
61.1 The old textcomp package code	320
61.1.1 Supporting oldstyle digits	328
61.1.2 Subset encoding defaults	329
 y ltpageno.dtx	 331
62 Page Numbering	331
 z ltxref.dtx	 332
63 Cross Referencing	332
63.1 Cross Referencing	332
 A ltmiscen.dtx	 336
64 Miscellaneous Environments	336
64.1 Environments	336
64.2 Center, Flushright, Flushleft	344
64.3 Verbatim	346
 B ltmath.dtx	 351
65 Math setup	351
65.1 Math commands based on plain T _E X	351
65.1.1 The log-like functions	351
65.1.2 Biggggg	352
65.1.3 The UNSORTED Rest	352
65.2 Math Environments	357
65.3 External options to the standard document classes	360
65.3.1 Left equation numbering	360
65.3.2 Flush left equations	361
 C ltlists.dtx	 364
66 List, and related environments	364
66.1 List and Trivlist	365
66.2 Vertical Spacing (skips)	366
66.3 Penalties	366
66.4 Horizontal Spacing (dimens)	366
66.5 Default Values	367
66.6 Itemize and Enumerate	377
 D ltboxes.dtx	 379

67 L^AT_EX Box commands	379
67.1 Some low-level constructs	391
E lttab.dtx	392
68 Tabbing, Tabular and Array Environments	392
68.1 tabbing	392
68.2 array and tabular environments	400
F ltpictur.dtx	414
69 Picture Mode	414
69.1 Curves	434
G ltthm.dtx	437
70 Theorem Environments	437
H ltsect.dtx	441
71 Sectioning Commands	441
71.1 The Title	441
71.2 Sectioning	442
71.2.1 Initializations	448
71.3 Table of Contents etc.	449
71.3.1 Convention	449
71.3.2 Commands	449
I ltfloat.dtx	453
72 Floats	453
72.1 Floating Environments	453
72.2 Footnotes	466
J ltidxglo.dtx	472
73 Index and Glossary Generation	472
K ltbibl.dtx	474
74 Bibliography Generation	474
74.1 Default definitions	477
L ltpage.dtx	478

75 Page styles and related commands	478
75.1 Page Style Commands	478
75.2 How a page style makes running heads and feet	478
75.3 marking conventions	478
 M ltoutput.dtx	 481
76 Output Routine	481
76.1 Floats	481
76.1.1 Kludgeins	536
76.1.2 Float control	537
76.1.3 Float placement parameters	549
 N ltclass.dtx	 553
77 Introduction	553
78 User interface	553
78.1 Option processing	554
79 Class and Package interface	555
79.1 Class name and version	555
79.2 Package name and version	555
79.3 Requiring other packages	555
79.4 Declaring new options	556
79.5 Safe Input Macros	557
80 Implementation	557
80.1 Hooks	570
80.2 Providing shipment	571
81 Package/class rollback mechanism	577
82 After Preamble	585
 O lthyphen.dtx	 586
 P ltluatex.dtx	 588
83 Overview	588
84 Core T_EX functionality	588
85 Plain T_EX interface	589

86 Lua functionality	589
86.1 Allocators in Lua	589
86.2 Lua access to T _E X register numbers	590
86.3 Module utilities	591
86.4 Callback management	591
87 Implementation	592
87.1 Minimum LuaT _E X version	592
87.2 Older L ^A T _E X/Plain T _E X setup	592
87.2.1 Fixes to <code>etex.src/etex.sty</code>	593
87.2.2 luatex specific settings	593
87.3 Attributes	594
87.4 Category code tables	594
87.5 Named Lua functions	596
87.6 Custom whatsits	597
87.7 Lua bytecode registers	597
87.8 Lua chunk registers	597
87.9 Lua loader	597
87.10 Lua module preliminaries	599
87.11 Lua module utilities	599
87.11.1 Module tracking	599
87.11.2 Module messages	600
87.12 Accessing register numbers from Lua	601
87.13 Attribute allocation	602
87.14 Custom whatsit allocation	602
87.15 Bytecode register allocation	603
87.16 Lua chunk name allocation	603
87.17 Lua function allocation	604
87.18 Lua callback management	604
87.18.1 Housekeeping	604
87.18.2 Handlers	606
87.18.3 Public functions for callback management	608
Q ltfinal.dtx	613
88 Final settings	613
88.1 Debugging	613
88.2 Typesetting parameters	613
88.3 Lccodes for hyphenation	615
88.4 Hyphenation	618
88.5 Font loading	619
88.6 Input encoding	619
88.7 Lccodes and uccodes	623
88.8 Applying Patch files	625
88.9 Loading further extensions	626
88.10 Freeing Memory	626
88.11 Initialise file list	627
88.12 Some last minute initializations ...	627
88.13 Do some temporary work for pre-release	627

88.14 Dumping the format	627
Change History	628
Index	691

File a

ltdirchk.dtx

1 L^AT_EX 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_EX 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_EX 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^AT_EX 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 , $3.0 \leq v \leq 3.14$

undefined otherwise.

However these values are accurate enough for L^AT_EX to take appropriate action for these old T_EXs.

If your T_EX is older than version 3.141, then you should define `\@TeXversion` (using `\def`) to be the version number. If you do not do this¹, L^AT_EX will not work around a bug in old T_EX 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_EX, 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}
```

¹Actually if your T_EX is really old, version 2, L^AT_EX 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 Lua_{TEX} is in use the extensions and other new primitives have to be activated: this is done as early as possible. Older versions of Lua_{TEX} 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 \ifinitex, latexrelease\fi
53 \ifdircheck\fi
54 \ifinitex\fi
55 \fi
56 \fi

```

A test can now be made for eTeX.

```

57 \ifinitex\ifx\etexversion\undefined
58 \ifinitex\errmessage
59 \ifinitex{LaTeX requires e-TeX}
60 \ifinitex\expandafter\endinput
61 \ifinitex\fi

```

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

```

62 \catcode'\#=6
63 \catcode'\^=7
64 \chardef\active=13
65 \catcode'\@=11
66 \countdef\count@=255
67 \let\bgroup={ \let\egroup=}
68 \ifx@@input\@undefined\let@@input\input\fi
69 \ifx@@end\@undefined\let@@end\end\fi
70 \chardef\@inputcheck0
71 \chardef\sixt@@n=16
72 \newlinechar'\^^J
73 \def\typeout{\immediate\write17}
74 \def\dospecials{\do\ \do\\\do\{\do\}\do\$\do\&%
75 \do\#\do\^{\do\_ \do\% \do\~}
76 \def\@makeother#1{\catcode'#1=12\relax}
77 \def\space{ }
78 \def\@tempwafalse{\let\if@tempwa\iffalse}
79 \def\@tempwattrue{\let\if@tempwa\iftrue}
80 \let\if@tempwa\iffalse
81 \def\loop#1\repeat{\def\iterate{#1\relax\expandafter\iterate\fi}%
82 \iterate \let\iterate\relax}
83 \let\repeat\fi
84 \ifinitex\fi

```

2.2 Some bits of 2e

```

85 \if2kernel
86 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}
87 \long\def\@firstoftwo#1#2{#1}
88 \long\def\@secondoftwo#1#2{#2}

```

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

```

89 \def\ProvidesFile#1{%
90 \begingroup
91 \catcode'\ 10 %
92 \ifnum \endlinechar<256 %
93 \ifnum \endlinechar>\m@ne
94 \catcode\endlinechar 10 %
95 \fi
96 \fi
97 \@makeother\/%

```

```

98     \ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]
99 \def\@providesfile#1[#2]{%
100     \wlog{File: #1 #2}%
101     \@addtofilelist{ #2}%
102     \endgroup}
103 \long\def\@addtofilelist#1{}
104 \def\@empty{}
105 \catcode'\%=12
106 \def\@percentchar{%}
107 \catcode'\%=14
108 \let\@currdir\@undefined
109 \let\input@path\@undefined
110 \let\filename@parse\@undefined

\strip@prefix

111 \def\strip@prefix#1>{}
112 </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.

```

113 <docstrip>
114 \openin15=texsys.cfg
115 \ifeof15
116 \typeout{** Writing a default texsys.cfg}
117 \immediate\openout15=texsys.cfg
118 \begingroup
119 \catcode'\^M\active%
120 \let^M\par%
121 \def\reserved@a#1^M{%
122     \def\reserved@b{#1}%
123     \ifx\reserved@b\reserved@c\endgroup\else%
124         \immediate\write15{#1}%
125         \expandafter\reserved@a\fi}%
126 \def\reserved@d#1START^M{\let\do\@makeother\dospecials\reserved@a}%
127 \catcode'\%=12
128 \def\reserved@c{END}
129 \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.

`\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 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.

```
130 %\def\@currdir{./}
131 %\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 `/`.

```
132 % \def\@currdir{./}
133 % \def\input@path{%
134 %   {/usr/local/lib/tex/inputs/distrib/}%
135 %   {/usr/local/lib/tex/inputs/contrib/}%
136 %   {/usr/local/lib/tex/inputs/local/}%
137 % }
```

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.

```
138 % \def\@currdir{./}
139 % \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.

```
140 % \def\@currdir{./}
141 % \def\input@path{%
142 %   {c:/tex/inputs/distrib/}%
143 %   {c:/tex/inputs/contrib/}%
144 %   {c:/tex/inputs/local/}%
145 % }
```

3.6 VMS (DECUS T_EX, 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.

```
146 % \def\@currdir{[] }
147 % \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:

```
148 % \def\@currdir{[]}  
149 % \def\input@path{%  
150 %   {tex_inputs:}%  
151 %   {SOMEDISK:[SOME.TEX.DIRECTORY]}%  
152 % }
```

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.

```
153 % \def\@currdir{:}  
154 % \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.

```
155 % \def\@currdir{:}  
156 % \def\input@path{%  
157 %   {Hard-Disk:Applications:TeX:TeX-inputs:}%  
158 %   {Hard-Disk:Applications:TeX:My-inputs:}%  
159 % }
```

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:

```
160 % \def\@dir#1#2 {%  
161 %   \@d@r{#1}#2..\@nil}  
162 % \def\@d@r#1#2.#3.#4\@nil{%  
163 %   <\ifx\@dir#1\@dir\else#1\ifx\@dir#3\@dir\else.\fi\fi#3>#2 }  
164 %  
165 % \def\@currdir{\@dir{}}  
166 % \def\input@path{%  
167 %   {\@dir{area.one}}%  
168 %   {\@dir{area.two}}%  
169 % }  
  
END  
  
170 \immediate\closeout15
```

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

```
171 \else
172 \typeout{** Using the existing texsys.cfg}
173 \closein15
174 \input texsys.cfg
175 \fi
176 \docstrip
```

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

```
177 \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.

```
178 \begingroup
179 \count@ \time
180 \divide \count@ 60
181 \count2 = - \count@
182 \multiply \count2 60
183 \advance \count2 \time
```

`\today` The current date and time stamp.

```
184 \edef \today {%
185   \the year / \two@digits{\the month} / \two@digits{\the day} : %
186   \two@digits{\the count@} : \two@digits{\the count2}}
```

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

```
187 \immediate \openout15 = texsys.aux
188 \immediate \write15 {\today ^^J}
189 \immediate \closeout15 %
```

`#1` is the file to try, `#2` is what to do on success, `#3` on failure. Note that this definition is overwritten later on again!

```
190 \def \IfFileExists#1#2#3{%
191   \openin \@inputcheck#1 %
192   \ifeof \@inputcheck
193     #3 \relax
194   \else
195     \read \@inputcheck to \reserved@a
196     \ifx \reserved@a \today
197       \typeout{#1 found} #2 \relax
198     \else
199       \typeout{BAD: old file \reserved@a (should be \today)} %
200       #3 \relax
201     \fi
202   \fi
203   \closein \@inputcheck}
```

```
204 \endlinechar=-1
```

If `\@currdir` has not been pre-defined in `texsys.cfg` then test for UNIX, VMS and Oz-TeX-Mac. syntax.

```
205 \ifx\@currdir\undefined
206   \IfFileExists{./texsys.aux}{\gdef\@currdir{.}}%
207   {\IfFileExists{[]texsys.aux}{\gdef\@currdir{[]}}%
208    {\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.

```
209 \ifx\@currdir\undefined
210   \global\let\@currdir\empty
211   \typeout{^^J^^J%
212     !! No syntax for the current directory could be found^^J%
213   }%
214 \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^AT_EX 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.

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

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

```
223 \fi

224 \immediate\closeout15 %
225 \endgroup

226 \typeout{^^J^^J%
227   \noexpand\@currdir set to:
228   \expandafter\strip@prefix\meaning\@currdir.^^J%
229 }
```

Stop here if the file is being used unstripped.

```
230 \docstrip
231 \relax\endinput
232 \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^AT_EX 2_ε 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^AT_EX 2_ε. 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.
233   \typeout{^^J%
234     Assuming \noexpand\openin and \noexpand\input^^J%
235     \ifx\input@path\undefined
\input@path has not been pre-defined.
236       have the same search path.^^J%
237     \else
\input@path has been defined in texsys.cfg.
238       have different search paths.^^J%
239       LaTeX will use the path specified by \noexpand\input@path:^^J%
240     \fi
241   }
```

6 Filename Parsing

```
\filename@parse Split a filename into its components.
242 \ifx\filename@parse\undefined
243   \def\reserved@a{.}\ifx\currdir\reserved@a
\filename@parse was not specified in texsys.cfg, but \@currdir looks like
UNIX...
244   \typeout{^^JDefining UNIX/DOS style filename parser.^^J}
245   \def\filename@parse#1{%
246     \let\filename@area\empty
247     \expandafter\filename@path#1/\}
      Search for the last /.
248     \def\filename@path#1/#2\{%
249       \ifx\#2\%
250         \def\reserved@a{\filename@simple#1.\}%
251       \else
252         \edef\filename@area{\filename@area#1/}%
253         \def\reserved@a{\filename@path#2\}%
254       \fi
255       \reserved@a}
256 \else\def\reserved@a{[]}\ifx\currdir\reserved@a
\filename@parse was not specified in texsys.cfg, but \@currdir looks like
VMS...
257   \typeout{^^JDefining VMS style filename parser.^^J}
258   \def\filename@parse#1{%
259     \let\filename@area\empty
260     \expandafter\filename@path#1/\}
```

Search for the last].

```

261 \def\filename@path#1:#2\\{%
262 \ifx\\#2\\%
263 \def\reserved@a{\filename@simple#1.\\}%
264 \else
265 \edef\filename@area{\filename@area#1}%
266 \def\reserved@a{\filename@path#2\\}%
267 \fi
268 \reserved@a}
269 \else\def\reserved@a{:}\ifx\@currdir\reserved@a

```

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

```

270 \typeout{^^JDefining Mac style filename parser.^^J}
271 \def\filename@parse#1{%
272 \let\filename@area\@empty
273 \expandafter\filename@path#1:}

```

Search for the last :.

```

274 \def\filename@path#1:#2\\{%
275 \ifx\\#2\\%
276 \def\reserved@a{\filename@simple#1.\\}%
277 \else
278 \edef\filename@area{\filename@area#1:}%
279 \def\reserved@a{\filename@path#2\\}%
280 \fi
281 \reserved@a}
282 \else

```

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

```

283 \typeout{^^JDefining generic filename parser.^^J}
284 \def\filename@parse#1{%
285 \let\filename@area\@empty
286 \expandafter\filename@simple#1.}
287 \fi\fi\fi

```

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

```

288 </dircheck>
289 <{*dircheck, latexrelease>
290 <[latexrelease]\IncludeInRelease{2019/10/01}{\filename@simple}
291 <[latexrelease] {Final dot for extension}%
292 \def\filename@simple#1.#2\\{%
293 \ifx\\#2\\%
294 \let\filename@ext\relax
295 \edef\filename@base{#1}%
296 \else
297 \filename@dots{#1}#2\\%
298 \fi}
299 \def\filename@dots#1#2.#3\\{%
300 \ifx\\#3\\%
301 \def\filename@ext{#2}%
302 \edef\filename@base{#1}%

```

```

303 \else
304   \filename@dots{#1.#2}#3\\%
305 \fi}
306 \latexrelease\EndIncludeInRelease
307 \latexrelease\IncludeInRelease{0000/00/00}{\filename@simple}
308 \latexrelease\{Final dot for extension}%
309 \latexrelease\def\filename@simple#1.#2\\{%
310 \latexrelease\ifx\#2\\%
311 \latexrelease\let\filename@ext\relax
312 \latexrelease\else
313 \latexrelease\edef\filename@ext{\filename@dot#2\\}%
314 \latexrelease\fi
315 \latexrelease\edef\filename@base{#1}}
316 \latexrelease\EndIncludeInRelease
317 \</dircheck, latexrelease>
318 \<(*dircheck>

    Remove a final dot, added earlier.
319 \def\filename@dot#1.\\{#1}
320 \else

Otherwise, \filename@parse was specified in texsys.cfg.
321 \typeout{^^J^^J%
322 \noexpand\filename@parse was defined in texsys.cfg:^^J%
323 \expandafter\strip@prefix\meaning\filename@parse.^^J%
324 }
325 \fi

```

7 T_EX Versions

\TeXversion T_EX versions older than 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_EX3.14 would be detected as T_EX3, but L^AT_EX only needs to take account of T_EX's older than 3, or between 3 and 3.14.

```

326 \ifx\TeXversion\undefined
327 \ifx\undefined\inputlineno
328 \def\TeXversion{2}
329 \else
330 {\catcode'\^^J=\active
331 \def\reserved@a#1#2\@@{\if#1\string~3\fi}
332 \edef\reserved@a{\expandafter\reserved@a\string^^J\@@}
333 \ifx\reserved@a\@empty\else\gdef\TeXversion{3}\fi}
334 \fi
335 \fi

336 \</dircheck>

```

8 lt_xcheck.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_EX

L^AT_EX includes almost all of the functionality of Knuth's original 'Basic Macros'. That is, the plain T_EX format described in Appendix B of the T_EXBook. 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^AT_EX font definitions are done using NFSS2 so none of PLAIN's font definitions are in L^AT_EX.

L^AT_EX has its own tabbing environment, so PLAIN's is disabled.

L^AT_EX uses its own output routine, so most of the plain one was removed.

```
1 (*2ekernel)
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 \def^~L{\par}% ascii form-feed is \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_EX 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.

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

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

End of historical L^AT_EX 2.09 comments.

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
\dimen@ 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
\newdimen \foo will be defined (with \countdef) to be the next counter.
\newskip To find out which counter \foo is, you can look at \allocationnumber.
\newmuskip Since there's no \boxdef command, \chardef is used to define a \newbox,
\newbox \newinsert, \newfam, and so on.
\newtoks
\newread
\newwrite
\newfam
\newlanguage
```

L^AT_EX 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 /2ekernel>
48 /*2ekernel | 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}

```

Skip `\write18` due to its traditional use as a shell-escape.

```

65 \ifx\directlua\@undefined
66   \def\newwrite {\e@alloc\write \chardef{\count17}\m@ne\sixt@@n}
67 \else
68   \def\newwrite {\e@alloc\write
69                 {\ifnum\allocationnumber=18
70                  \advance\count17\@ne
71                  \allocationnumber\count17 %
72                  \fi
73                  \global\chardef}%
74                 {\count17}%
75                 \m@ne
76                 {128}}
77 \fi

78 \def\new@mathgroup
79   {\e@alloc\mathgroup\chardef{\count18}\m@ne\e@mathgroup@top}
80 \let\newfam\new@mathgroup

81 \ifx\directlua\@undefined
82   \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne\@ccclvi}
83 \else
84   \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne{16384}}
85 \fi
86 /2ekernel | latexrelease>

87 (latexrelease)\EndIncludeInRelease
88 (latexrelease)\IncludeInRelease{0000/00/00}%
89 (latexrelease)          {\newcount}{Extended Allocation}%
90 (latexrelease)\def\newcount{\alloc@0\count\countdef\insc@unt}
91 (latexrelease)\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}

```

```

92 <latexrelease>\def\newskip{\alloc@2\skip\skipdef\insc@unt}
93 <latexrelease>\def\newmuskip{\alloc@3\muskip\muskipdef\@ccclvi}
94 <latexrelease>\def\newbox{\alloc@4\box\chardef\insc@unt}
95 <latexrelease>\def\newtoks{\alloc@5\toks\toksdef\@ccclvi}
96 <latexrelease>\def\newread{\alloc@6\read\chardef\sixt@@n}
97 <latexrelease>\def\newwrite{\alloc@7\write\chardef\sixt@@n}
98 <latexrelease>\def\new@mathgroup{\alloc@8\fam\chardef\sixt@@n}
99 <latexrelease>\def\newlanguage{\alloc@9\language\chardef\@ccclvi}
100 <latexrelease>\let\newfam\new@mathgroup
101 <latexrelease>\EndIncludeInRelease

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

102 <*2kernel | latexrelease>
103 <latexrelease>\IncludeInRelease{2015/01/01}%
104 <latexrelease>{\@alloc@chardef}{Extended Allocation}%

105 \ifx\directlua\@undefined
106 \ifx\widowpenalties\@undefined

classic tex has 28 registers.

107 \mathchardef\@alloc@top=255
108 \let\@alloc@chardef\chardef
109 \else

etex and xetex have 215 registers.

110 \mathchardef\@alloc@top=32767
111 \let\@alloc@chardef\mathchardef
112 \fi
113 \else

luatex has 216 registers.

114 \chardef\@alloc@top=65535
115 \let\@alloc@chardef\chardef
116 \fi

117 </2kernel | latexrelease>
118 <latexrelease>\EndIncludeInRelease
119 <latexrelease>\IncludeInRelease{0000/00/00}%
120 <latexrelease>{\@alloc@chardef}{Extended Allocation}%
121 <latexrelease>\let\@alloc@top\@undefined
122 <latexrelease>\let\@alloc@chardef\@undefined
123 <latexrelease>\EndIncludeInRelease

\@mathgroup@top The upper limit of extended math groups (\fam) 16 in classic TEX and e-TEX, but
256 in Unicode TeX variants.

124 <*2kernel | latexrelease>
125 <latexrelease>\IncludeInRelease{2015/01/01}%
126 <latexrelease>{\@mathgroup@top}{Extended Allocation}%

127 \ifx\Umathcode\@undefined

classic and e tex have 16 fam (0–15).

128 \chardef\@mathgroup@top=16
129 \else

```

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

```

130 \chardef\@mathgroup@top=256
131 \fi
132 </2ekernel | latexrelease>
133 <latexrelease>\EndIncludeInRelease
134 <latexrelease>\IncludeInRelease{0000/00/00}%
135 <latexrelease> \{\@mathgroup@top\}{Extended Allocation}%
136 <latexrelease>\let\@mathgroup@top\@undefined
137 <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.
138 <*2ekernel | latexrelease>
139 <latexrelease>\IncludeInRelease{2015/01/01}\{\@alloc\}{Extended Allocation}%
140 \def\@alloc#1#2#3#4#5#6{%
141 \global\advance#3\@ne
142 \e@ch@ck{#3}{#4}{#5}#1%
143 \allocationnumber#3\relax
144 \global#2#6\allocationnumber
145 \wlog{\string#6=\string#1\the\allocationnumber}}%
146 </2ekernel | latexrelease>
147 <latexrelease>\EndIncludeInRelease
148 <latexrelease>\IncludeInRelease{0000/00/00}\{\@alloc\}{Extended Allocation}%
149 <latexrelease>\let\@alloc\@undefined
150 <latexrelease>\EndIncludeInRelease
151 <*2ekernel>

```

\@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`.

```

152 </2ekernel>
153 <*2ekernel | latexrelease>
154 <latexrelease>\IncludeInRelease{2015/10/01}
155 <latexrelease> \{\@ch@ck\}{Extended Allocation (checking)}%
156 \gdef\@ch@ck#1#2#3#4{%
157 \ifnum#1<#2\else
158 \ifnum#1=#2\relax
159 \global#1\@cclvi
160 \ifx\count#4\global\advance#1 10 \fi
161 \fi

```

Check we are below the extended limit.

```

162 \ifnum#1<#3\relax
163 \else
164 \errmessage{No room for a new \string#4}%
165 \fi
166 \fi}%
167 \<latexrelease>\EndIncludeInRelease
168 \<latexrelease>\IncludeInRelease{2015/01/01}%
169 \<latexrelease>{\e@ch@ck}{Extended Allocation (checking)}%
170 \<latexrelease>\gdef\e@ch@ck#1#2#3#4{%
171 \<latexrelease> \ifnum#1<#2\else
172 \<latexrelease> \ifnum#1=#2\relax
173 \<latexrelease> #1\@cclvi
174 \<latexrelease> \ifx\count#4\advance#1 10 \fi
175 \<latexrelease> \fi
176 \<latexrelease> \ifnum#1<#3\relax
177 \<latexrelease> \else
178 \<latexrelease> \errmessage{No room for a new #4}%
179 \<latexrelease> \fi
180 \<latexrelease> \fi}%
181 \<latexrelease>\EndIncludeInRelease
182 \<latexrelease>\IncludeInRelease{0000/00/00}%
183 \<latexrelease>{\e@ch@ck}{Extended Allocation (checking)}%
184 \<latexrelease>\let\e@ch@ck\@undefined
185 \<latexrelease>\EndIncludeInRelease

186 \<latexrelease>\IncludeInRelease{2015/01/01}%
187 \<latexrelease>{\extrafloats}{Extra floats}%

188 \let\float@count\e@alloc@top

```

\extrafloats

```

189 \ifx\numexpr\@undefined
In classic TeX use \newinsert to allocate float boxes.
190 \def\extrafloats#1{%
191 \count@#1\relax
192 \ifnum\count@>\z@
193 \newinsert\reserved@a
194 \global\expandafter\chardef
195 \csname bx@the\allocationnumber\endcsname\allocationnumber
196 \@cons\@freelist{\csname bx@the\allocationnumber\endcsname}%
197 \advance\count@\m@ne
198 \expandafter\extrafloats
199 \expandafter\count@
200 \fi
201 }%

202 \else
In e-tex take float boxes from the top of the extended range.
203 \def\extrafloats#1{%
204 \ifnum#1>\z@
205 \count@\numexpr\float@count-1\relax
206 \ch@ck0\count@\count
207 \ch@ck1\count@\dimen

```



```

208 \ch@ck2\count@\skip
209 \ch@ck4\count@\box
210 \global\@alloc@chardef\float@count\count@
211 \global\expandafter\@alloc@chardef
212 \csname bx@\the\float@count\endcsname\float@count
213 \@cons\@freelist{\csname bx@\the\float@count\endcsname}%
214 \expandafter
215 \extrafloats\expandafter{\numexpr#1-1\relax}%
216 \fi}%
217 \fi

218 /2ekernel | latexrelease)
219 (latexrelease)\EndIncludeInRelease
220 (latexrelease)\IncludeInRelease{0000/00/00}%
221 (latexrelease) {\extrafloats}{Extra floats}%
222 (latexrelease)\let\float@count\@undefined
223 (latexrelease)\let\extrafloats\@undefined
224 (latexrelease)\EndIncludeInRelease
225 (*2ekernel)

\alloc@

226 \def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
227 \ch@ck#1#4#2%
228 \allocationnumber\count1#1%
229 \global#3#5\allocationnumber
230 \wlog{\string#5=\string#2\the\allocationnumber}}

\newinsert

231 /2ekernel)
232 (*2ekernel | latexrelease)
233 (latexrelease)\IncludeInRelease{2015/10/01}
234 (latexrelease) {\newinsert}{Extended \newinsert}%

235 \ifx\numexpr\@undefined
If e-TeX is not available use the original plain TeX definition of \newinsert.
236 \def\newinsert#1{\global\advance\insc@unt \m@ne
237 \ch@ck0\insc@unt\count
238 \ch@ck1\insc@unt\dimen
239 \ch@ck2\insc@unt\skip
240 \ch@ck4\insc@unt\box
241 \allocationnumber\insc@unt
242 \global\chardef#1\allocationnumber
243 \wlog{\string#1=\string\insert\the\allocationnumber}}

244 \else
The highest register allowed with \insert.
245 \ifx\directlua\@undefined
246 \chardef\@insert@top255
247 \else
248 \chardef\@insert@top\@alloc@top
249 \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.

```

```

250 \def\newinsert#1{%
251 \@tempswafalse
252 \global\advance\insc@unt\m@ne
253 \ifnum\count10<\insc@unt
254 \ifnum\count11<\insc@unt
255 \ifnum\count12<\insc@unt
256 \ifnum\count14<\insc@unt
257 \@tempswatrue
258 \fi\fi\fi\fi
259 \if@tempswa
260 \allocationnumber\insc@unt
261 \else
262 \global\advance\insc@unt\@ne
263 \extrafloats\@ne
264 \@next\@currbox\@freelist
265 {\ifnum\@currbox<\e@insert@top
266 \allocationnumber\@currbox
267 \else
268 \ch@ck0\m@ne\insert
269 \fi}%
270 {\ch@ck0\m@ne\insert}%
271 \fi
272 \global\chardef#1\allocationnumber
273 \wlog{\string#1=\string\insert\the\allocationnumber}%
274 }

275 \fi
276 </2ekernel | latexrelease>

277 <latexrelease>\EndIncludeInRelease
278 <latexrelease>\IncludeInRelease{0000/00/00}%
279 <latexrelease> {\newinsert}{Extended \newinsert}%
280 <latexrelease>\let\e@insert@top\@undefined
281 <latexrelease>\def\newinsert#1{\global\advance\insc@unt \m@ne
282 <latexrelease> \ch@ck0\insc@unt\count
283 <latexrelease> \ch@ck1\insc@unt\dimen
284 <latexrelease> \ch@ck2\insc@unt\skip
285 <latexrelease> \ch@ck4\insc@unt\box
286 <latexrelease> \allocationnumber\insc@unt
287 <latexrelease> \global\chardef#1\allocationnumber
288 <latexrelease> \wlog{\string#1=\string\insert\the\allocationnumber}}
289 <latexrelease>\EndIncludeInRelease
290 <*2ekernel>

\ch@ck

291 \gdef\ch@ck#1#2#3{%
292 \ifnum\count1#1<#2\else
293 \errmessage{No room for a new #3}%
294 \fi}

\newhelp

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

\maxdimen Here are some examples of allocation.
\hideskip

```

```

296 \newdimen\maxdimen \maxdimen=16383.99999pt % the largest legal <dimen>
297 \newskip\hideskip \hideskip=-1000pt plus 1fill % negative but can grow

\p@
\z@ 298 \newdimen\p@ \p@=1pt % this saves macro space and time
\z@skip 299 \newdimen\z@ \z@=0pt % can be used both for 0pt and 0
\voidb@x 300 \newskip\z@skip \z@skip=0pt plus0pt minus0pt
301 \newbox\voidb@x % permanently void box register

Assign initial values to TeX's parameters
302 \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.
Historical ETeX 2.09 comments (not necessarily accurate any more):
303 \pretolerance=100
304 \tolerance=200 % INITEX sets this to 10000
305 \hbadness=1000
306 \vbadness=1000
307 \linepenalty=10
308 \hyphenpenalty=50
309 \exhyphenpenalty=50
310 \binoppenalty=700
311 \relpenalty=500
312 \clubpenalty=150
313 \widowpenalty=150
314 \displaywidowpenalty=50
315 \brokenpenalty=100
316 \predisplaypenalty=10000
\postdisplaypenalty=0
\interlinepenalty=0
\floatingpenalty=0, set during \insert
\outputpenalty=0, set before TeX enters \output
317 \doublehyphendemerits=10000
318 \finalhyphendemerits=5000
319 \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
320 \tracinglostchars=1
\tracingcommands=0
\tracingrestores=0
\language=0
321 \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

322 \defaultthyphenchar='\-
323 \defaultskewchar=-1

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

324 \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

```

End of historical L^AT_EX 2.09 comments.

In L^AT_EX we don't want box information in the transcript unless we do a full tracing.

```

325 \showboxbreadth=-1
326 \showboxdepth=-1
327 \errorcontextlines=-1

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

```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```

\lineskiplimit=0pt, changed by \normalbaselines

334 \delimitershortfall=5pt
335 \nulldelimiterspace=1.2pt
336 \scriptspace=0.5pt

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

337 \parindent=20pt

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

\baselineskip=0pt, changed by \normalbaselines
\lineskip=0pt, changed by \normalbaselines

338 \parskip=0pt plus 1pt

```

```

339 \abovedisplayskip=12pt plus 3pt minus 9pt
340 \abovedisplayshortskip=0pt plus 3pt
341 \belowdisplayskip=12pt plus 3pt minus 9pt
342 \belowdisplayshortskip=7pt plus 3pt minus 4pt

\leftskip=0pt
\rightskip=0pt

343 \topskip=10pt
344 \splittopskip=10pt

\tabskip=0pt
\spaceskip=0pt
\xspaceskip=0pt

345 \parfillskip=0pt plus 1fil
End of historical LATEX 2.09 comments.

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

\interfootlinepenalty
349 \newcount\interfootnotelinepenalty \interfootnotelinepenalty=100

Definitions for preloaded fonts

\magstephalf
\magstep 350 \def\magstephalf{1095 }
351 \def\magstep#1{\ifcase#1 \@m\or 1200\or 1440\or 1728\or
352 2074\or 2488\fi\relax}

Macros for setting ordinary text

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

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

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

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

\lbrack
\rbrack 363 \def\lbrack{[ }
364 \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.
365 \def \aa {\r a}
366 \def \AA {\r A}

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

\space
369 \def\space{ }

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

\null
371 \def\null{\hbox{}}

\bgroup
\egroup 372 \let\bgroup={
373 \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{...
374 {\catcode'\^^M=\active % these lines must end with %
375 \gdef\obeylines{\catcode'\^^M\active \let^^M\par}%
376 \global\let^^M\par} % this is in case ^^M appears in a \write
377 \def\obeyspaces{\catcode'\ \active}
378 {\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).
379 \long\def \loop #1\repeat{%
380 \def\iterate{#1\relax % Extra \relax
381 \expandafter\iterate\fi
382 }%
383 \iterate
384 \let\iterate\relax
385 }

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

LATEX defines \smallskip, etc. in lt-space.dtx.

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

```

```

\vglue
\hglue 390 \def\vglue{\afterassignment\vgl@\skip@=}
391 \def\vgl@{\par \dimen@\prevdepth \hrule \@height\z@
392 \nobreak\vskip\skip@ \prevdepth\dimen@}
393 \def\hglue{\afterassignment\hgl@\skip@=}
394 \def\hgl@{\leavevmode \count@\spacefactor \vrule \@width\z@
395 \nobreak\hskip\skip@ \spacefactor\count@}

LATEX defines ~ in ltdefns.dtx.

\slash This generates a / acting a bit like - but still allows hyphenation in the word part
preceding it (but not after).
396 \def\slash{/\penalty\exhyphenpenalty}

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

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

\eject Define \eject as in plain TEX but define \supereject only in the compatibility
file.
402 \def\eject{\par\break}

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

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

\m@th
410 \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?).
411 \def\underbar#1{\underline{\sbox\tw@{#1}\dp\tw@\z@\box\tw@}}

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

\hidewidth For alignment entries that can stick out.
414 \def\hidewidth{\hskip\hideskip}

```

`\narrower`

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

L^AT_EX defines `\ae` and similar commands elsewhere.

```
418 \chardef\%='\'%
419 \chardef\&='&%
420 \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

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

`\mathhexbox`

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

`\ialign`

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

`\oalign`

`\o@lign`

`\ooalign`

```
424 \def\oalign#1{\leavevmode\topof\baselineskip\z@skip \lineskip.25ex%
425   \ialign{##\crrc#1\crrc}}
426 \def\o@lign{\lineskiplimit\z@ \oalign}
427 \def\ooalign{\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^AT_EX.

Since the version given here has been in use by L^AT_EX 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.

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

`\ltx@sh@ft`

This is the L^AT_EX 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`).

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

L^AT_EX change: the text commands such as `\d`, `\b`, `\c`, `\copyright`, `\TeX` are now defined elsewhere.

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

INITEX sets \sfcode x=1000 for all x, except that \sfcode‘X’=999 for upper-
case letters. The following changes are needed:
440 \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:
441 \def\showoverfull{\tracingonline\@ne}

\showoutput
\loggingoutput 442 \gdef\loggingoutput{\tracingoutput\@ne
443   \showboxbreadth\maxdimen\showboxdepth\maxdimen\errorstopmode}
444 \gdef\showoutput{\loggingoutput\showoverfull}
445 /2kernel)

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

```

```

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

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

```

```

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

527 \def\hideoutput{%
528   \tracingoutput\z@
529   \showboxbreadth\m@ne
530   \showboxdepth\m@ne
531   \tracingonline\m@ne
532 }%

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

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

540 <*2ekernel>
541 \nonfrenchspacing
542 </2ekernel>

```

File c

ltvers.dtx

10 Version Identification

First we identify the date and version number of this release of L^AT_EX, and set `\everyjob` so that it is printed at the start of every L^AT_EX run.

`\fmtname` A `\patch@level` of 0 or higher denotes an official public release. A negative value indicates a candidate release that is not distributed.

`\fmtversion` If we put code updates into the kernel that are supposed to go into the next release we set the `\patch@level` to -1 and the `\fmtversion` / `\latexreleaseversion` to the date of the next release (guessed, the real value is not so important and will get corrected when we make the release official).

`\patch@level` If the `\patch@level` is already at -1 we do nothing here and use the `\fmtversion` date for any new `\IncludeInRelease` line when we add further code.

Finally, if we do make a public release we either just set the `\patch@level` to zero (if our initial guess was good) or we also change the date and then have to additionally change to that date on all the `\IncludeInRelease` statements that used the “guessed” date.

```

1 (*2ekernel)
2 \def\fmtname{LaTeX2e}
3 \edef\fmtversion
4 (/2ekernel)
5 (latexrelease)\edef\latexreleaseversion
6 (*2ekernel| latexrelease)
7 {2020-02-01}
8 (/2ekernel| latexrelease)
9 (*2ekernel)
10 \def\patch@level{-3}

```

`\development@branch@name` For more fine grain control there is the possibility to name the current development branch. This is only used when the `\patch@level` is negative (i.e., a pre-release format) and is intended to help help us internally when we locally install a format out of some development branch.

```

11 \edef\development@branch@name{develop \the\year-\the\month-\the\day}

```

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:-)).

```

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

```



```

68 \latexrelease\newif\if@includeinrelease
69 \latexrelease\@includeinreleasefalse

70 \def\IncludeInRelease#1{%
71   \if@includeinrelease
72     \PackageError{latexrelease}{mis-matched IncludeInRelease}%
73       {There is an \string\EndIncludeRelease\space missing}%
74     \@includeinreleasefalse
75   \fi
76   \kernel@ifnextchar[%
77     {\@IncludeInRelease{#1}}
78     {\@IncludeInRelease{#1}[#1]}

    If a specific date has not been specified in latexrelease use ‘#1’.

79 \def\@IncludeInRelease#1[#2]{\@IncludeInRelease@se{#2}}

80 \def\@IncludeInRelease@se#1#2#3{%
81   \toks@{[#1] #3}%
82   \expandafter\ifx\csname\string#2+\@currname+IIR\endcsname\relax
83     \ifnum\expandafter\@parse@version#1//00\@nil
84       >\expandafter\@parse@version\fmtversion//00\@nil
85       \GenericInfo{}{Skipping: \the\toks@}%
86       \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
87     \else
88       \GenericInfo{}{Applying: \the\toks@}%
89       \@includeinreleasetrue
90       \expandafter\let\csname\string#2+\@currname+IIR\endcsname\@empty
91     \fi
92   \else
93     \GenericInfo{}{Already applied: \the\toks@}%
94     \expandafter\@gobble@IncludeInRelease
95   \fi
96 }

97 \def\EndIncludeInRelease{%
98   \if@includeinrelease
99     \@includeinreleasefalse
100 \else
101   \PackageError{latexrelease}{mis-matched EndIncludeInRelease}{}%
102 \fi}

103 \long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{%
104   \@includeinreleasefalse
105   \@check@IncludeInRelease#1\IncludeInRelease\@check@IncludeInRelease
106   \@end@check@IncludeInRelease}

107 \long\def\@check@IncludeInRelease#1\IncludeInRelease
108                                     #2#3\@end@check@IncludeInRelease{%
109   \ifx\@check@IncludeInRelease#2\else
110     \PackageError{latexrelease}{skipped IncludeInRelease for tag \string#2}{}%
111   \fi}

112 </2ekernel | latexrelease>

```

File d

ltxdefns.dtx

11 Definitions

This section contains commands used in defining other macros.

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

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 TeX 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` Save original primitive definition.

```
9 \let\@@hyph=\-
```

`\@@italiccorr` Save the original italic correction.

```
10 \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.

```
11 \def\@height{height} \def\@depth{depth} \def\@width{width}
12 \def\@minus{minus}
13 \def\@plus{plus}
```

`\hb@xt@` The next one is another 100 tokens worth.

```
14 \def\hb@xt@{\hbox to}
15 \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–\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’ \neq ‘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 <code>\FOO</code> to be a macro with <i>i</i> arguments (<i>i</i> = 0 if missing) having the definition <code>\TEXT</code> . Produces an error if <code>\FOO</code> already defined.
		Normally the command is defined to be <code>\long</code> (ie it may take multiple paragraphs in its argument). In the star-form, the command is not defined as <code>\long</code> and a blank line in any argument to the command would generate an error.
<code>\renewcommand</code>	<code>*{\FOO}[i]{\TEXT}</code>	
		Same as <code>\newcommand</code> , except it checks if <code>\FOO</code> already defined.
<code>\newenvironment</code>	<code>*{\FOO}[i]{\DEF1}{\DEF2}</code>	
		equivalent to: <code>\newcommand{\FOO}[i]{\DEF1} \def{\endFOO}{\DEF2}</code> (or the appropriate star forms).
<code>\renewenvironment</code>		Obvious companion to <code>\newenvironment</code> .
<code>\@cons</code>		: See description of <code>\output</code> routine.
<code>\@car</code>	<code>\@car T1 T2 ... Tn\@nil == T1</code>	(unexpanded)
<code>\@cdr</code>	<code>\@cdr T1 T2 ... Tn\@nil == T2 ... Tn</code>	(unexpanded)
<code>\typeout</code>	<code>{\message}</code>	
		Produces a warning message on the terminal.
<code>\typein</code>	<code>{\message}</code>	
		Types message, asks the user to type in a command, then executes it
<code>\typein</code>	<code>[\CS]{\MSG}</code>	
		Same as above, except defines <code>\CS</code> to be the input instead of executing it.
<code>\typein</code>		
	<code>16 \def\typein{%</code>	


```

17 \let\@typein\relax
18 \@testopt\@xtypein\@typein}

19 \ifx\directlua\@undefined

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

26 \else

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

34 \fi

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

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

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

\@car
\@cdr
38 \def\@car#1#2\@nil{#1}
39 \def\@cdr#1#2\@nil{#2}

\@carcube \@carcube T1 ... Tn\@nil = T1 T2 T3 ,  $n > 3$ 
40 \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.
41 \def\@preamblecmds{}
42 \def\@onlypreamble#1{%
43 \expandafter\gdef\expandafter\@preamblecmds\expandafter{%
44 \@preamblecmds\do#1}}
45 \@onlypreamble\@onlypreamble
46 \@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.
47 \def\@star@or@long#1{%
48 \@ifstar
49 {\let\l@ngrel@x\relax#1}%
50 {\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.

```
51 \let\l@ngrel@x\relax
```

`\newcommand` User level `\newcommand`.

```
52 \def\newcommand{\@star@or@long\new@command}
```

`\new@command`

```
53 \def\new@command#1{%
54   \@testopt{\@newcommand#1}0}
```

`\@newcommand` Handling arguments for `\newcommand`.

```
\@argdef 55 \def\@newcommand#1[#2]{%
\@xargdef 56   \kernel@ifnextchar [{\@xargdef#1[#2]}%
57               {\@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.

```
58 \long\def\@argdef#1[#2]#3{%
59   \@ifdefinable #1{\@yargdef#1\@ne{#2}{#3}}}
```

Handle the second optional argument.

```
60 \long\def\@xargdef#1[#2] [#3]#4{%
61   \@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`.

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

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

```
69   \expandafter\@yargdef
70   \csname\string#1\endcsname
71   \tw@
72   {#2}%
73   {#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.

```
74 \long\def\@testopt#1#2{%
75   \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 `\@x@protect`). 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.

```
76 \def\@protected@testopt#1{%
77   \ifx\protect\@typeset@protect
78     \expandafter\@testopt
79   \else
80     \@x@protect#1%
81   \fi}
```

`\@yargdef` These generate a primitive argument specification, from a L^AT_EX [*<digit>*] form; `\@yargd@f` in fact *<digit>* can be anything such that `\number <digit>` is single digit.

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^AT_EX 2.09 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`.

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

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

```

\@reargdef
    97 \long\def\@reargdef#1[#2]{%
    98   \@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.)
    99 \def\renewcommand{\@star@or@long\renew@command}

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

\@ifdefinable Test is user is allowed to define a command.
\@@ifdefinable
\@rc@ifdefinable
    107 \long\def\@ifdefinable #1#2{%
    108   \edef\reserved@a{\expandafter\@gobble\string #1}%
    109   \@ifundefined\reserved@a
    110     {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
    111      \ifx \reserved@b\@qend \@notdefinable\else
    112        \ifx \reserved@a\@qrelax \@notdefinable\else
    113          #2%
    114        \fi
    115      \fi}%
    116   \@notdefinable}
Saved definition of \@ifdefinable.
    117 \let\@@ifdefinable\@ifdefinable
Version of \@ifdefinable for use with \renewcommand. Does not do the check
this time, but restores the normal definition.
    118 \long\def\@rc@ifdefinable#1#2{%
    119   \let\@ifdefinable\@@ifdefinable
    120   #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.
    121 \def\newenvironment{\@star@or@long\new@environment}

\new@environment
    122 \def\new@environment#1{%
    123   \@testopt{\@newenva#1}0}

```

```

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

\@newenvb
126 \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.
127 \def\renewenvironment{\@star@or@long\renew@environment}

\renew@environment
128 \def\renew@environment#1{%
129   \@ifundefined{#1}%
130     {\@latex@error{Environment #1 undefined}\@ehc
131     }\relax
132   \expandafter\let\csname#1\endcsname\relax
133   \expandafter\let\csname end#1\endcsname\relax
134   \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.
135 \long\def\@newenv#1#2#3#4{%
136   \@ifundefined{#1}%
137     {\expandafter\let\csname#1\expandafter\endcsname
138       \csname end#1\endcsname}%
139     \relax
140   \expandafter\new@command
141     \csname #1\endcsname#2{#3}%
142     \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.
143 \def\newif#1{%
144   \count@\escapechar \escapechar\m@ne
145   \let#1\iffalse
146   \@if#1\iftrue
147   \@if#1\iffalse
148   \escapechar\count@}

\@if
149 \def\@if#1#2{%
150   \expandafter\def\csname\expandafter\@gobbletwo\string#1%
151     \expandafter\@gobbletwo\string#2\endcsname
152     {\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 imple-
mentation 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.

```
153 \def\providecommand{\@star@or@long\providecommand}
```

`\providecommand`

```
154 \def\providecommand#1{%
155   \begingroup
156   \escapechar\m@ne\xdef\@gtempa{\string#1}%
157   \endgroup
158   \expandafter\@ifundefined\@gtempa
159   {\def\reserved@a{\newcommand#1}}%
160   {\def\reserved@a{\renewcommand\reserved@a}}%
161   \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.

```
162 \def\CheckCommand{\@star@or@long\checkcommand}
```

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

```
163 \@onlypreamble\CheckCommand
```

`\checkcommand`

```
164 \def\checkcommand#1#2#{\@check@c#1{#2}}
165 \@onlypreamble\checkcommand
```

`\@check@c` `\CheckCommand` itself just grabs all the arguments we need, without actually looking 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`.

```
166 \long\def\@check@c#1#2#3{%
167   \expandafter\let\csname\string\reserved@a\endcsname\relax
168   \renewcommand\reserved@a#2{#3}%
169   \@ifundefined{\string\reserved@a}%
170   {\@check@eq#1\reserved@a}%
171   {\expandafter\@check@eq
172     \csname\string#1\expandafter\endcsname
173     \csname\string\reserved@a\endcsname}}
174 \@onlypreamble\@check@c
```

`\@check@eq` Complain if `\#1` and `\#2` are not `\ifx` equal.

```
175 \def\@check@eq#1#2{%
176   \ifx#1#2\else
177     \latex@warning@no@line
178       {Command \noexpand#1 has
179         changed.\MessageBreak
180         Check if current package is valid}%
181   \fi}
182 \@onlypreamble\@check@eq
```

```

\@gobble The \@gobble macro is used to get rid of its argument.
\@gobbletwo 183 \long\def \@gobble #1{}
\@gobblefour 184 \long\def \@gobbletwo #1#2{}
185 \long\def \@gobblefour #1#2#3#4{}

\@firstofone Some argument-grabbers.
\@firstoftwo 186 \long\def \@firstofone#1{#1}
\@secondoftwo 187 \long\def \@firstoftwo#1#2{#1}
188 \long\def \@secondoftwo#1#2{#2}

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

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

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

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

```

11.4 Robust commands and protect

Fragile and robust commands are one of the thornier issues in L^AT_EX's commands. Whilst typesetting documents, L^AT_EX makes use of many of T_EX'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^AT_EX, 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^AT_EX 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. More precisely, whenever the content of an `\edef` or `\xdef`

etc. can contain arbitrary user input not under the direct control of the programmer, one should use `\protected@edef` instead of `\edef`, etc., so that `\protect` has a suitable definition and the user input will not break if it contains fragile commands.

`\@unexpandable@protect`

```
194 \def\@unexpandable@protect{\noexpand\protect\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 `\protect` 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`.

```
195 \def\DeclareRobustCommand{\@star@or@long\declare@robustcommand}
196 \def\declare@robustcommand#1{%
197   \ifx#1\@undefined\else\ifx#1\relax\else
198     \@latex@info{Redefining \string#1}%
199   \fi\fi
200   \edef\reserved@a{\string#1}%
201   \def\reserved@b{#1}%
202   \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
203   \edef#1{%
204     \ifx\reserved@a\reserved@b
205       \noexpand\x@protect
206       \noexpand#1%
207     \fi
```



```

208     \noexpand\protect
209     \expandafter\noexpand\csname
210     \expandafter\@gobble\string#1 \endcsname
211 }%
212 \let\@ifdefinable\@rc@ifdefinable
213 \expandafter\new@command\csname
214     \expandafter\@gobble\string#1 \endcsname
215 }

\@x@protect
\@x@protect 216 \def\@x@protect#1{%
217     \ifx\protect\@typeset@protect\else
218         \@x@protect#1%
219     \fi
220 }
221 \def\@x@protect#1\fi#2#3{%
222     \fi\protect#1%
223 }

\@typeset@protect We set \@typeset@protect to \relax rather than \@empty to make sure that the
                    protection mechanism stops the look-ahead and expansion performed at the start
                    of \halign cells.
224 \let\@typeset@protect\relax

\set@display@protect These macros set \protect appropriately for typesetting or displaying.
\set@typeset@protect 225 \def\set@display@protect{\let\protect\string}
226 \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!
227 \def\protected@edef{%
228     \let\@protect\protect
229     \let\protect\@unexpandable@protect
230     \afterassignment\restore@protect
231     \edef
232 }
233 \def\protected@xdef{%
234     \let\@protect\protect
235     \let\protect\@unexpandable@protect
236     \afterassignment\restore@protect
237     \xdef
238 }
239 \def\unrestored@protected@xdef{%
240     \let\protect\@unexpandable@protect
241     \xdef
242 }
243 \def\restore@protect{\let\protect\@protect}

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

```

`\MakeRobust` This macro makes an existing fragile macro robust, but only if it hasn't been robust in the past, i.e., it checks for the existence of the macro `\<name>` and if that exists it assumes that `\<name>` is already robust. In that case either undefine the inner macro first or use `\DeclareRobustCommand` to define it in a robust way directly. We could probably test the top-level definition to have the right kind of structure, but this is somewhat problematical as we then have to distinguish between `\long` macros and others and also take into account that sometimes the top-level is deliberately done manually (like with `\begin`).

The macro firstly checks if the control sequence in question exists at all.

```

245 </2ekernel>
246 <latexrelease>\IncludeInRelease{2019/10/01}{\MakeRobust}{\MakeRobust}%
247 <*2ekernel | latexrelease>
248 \def\MakeRobust#1{%
249   \ifundefined{\expandafter\@gobble\string#1}{%
250     \latexerror{The control sequence '\string#1' is undefined!%
251       \MessageBreak There is nothing here to make robust}%
252     \@eha
253   }%

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.

254   {%
255     \ifundefined{\expandafter\@gobble\string#1\space}%
256     {%
257       \global\expandafter\let\csname
258         \expandafter\@gobble\string#1\space\endcsname=#1%
259       \edef\reserved@a{\string#1}%
260       \def\reserved@b{#1}%
261       \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
262       \xdef#1{%
263         \ifx\reserved@a\reserved@b
264           \noexpand\x@protect\noexpand#1%
265         \fi
266         \noexpand\protect\expandafter\noexpand
267         \csname\expandafter\@gobble\string#1\space\endcsname}%
268     }%
269     {\@latexinfo{The control sequence '\string#1' is already robust}}%
270   }%
271 }%
272 </2ekernel | latexrelease>
273 <latexrelease>\EndIncludeInRelease
274 <latexrelease>\IncludeInRelease{2015/01/01}{\MakeRobust}{\MakeRobust}%
275 <latexrelease>\def\MakeRobust#1{%
276 <latexrelease>  \ifundefined{\expandafter\@gobble\string#1}{%
277 <latexrelease>    \latexerror{The control sequence '\string#1' is undefined!%
278 <latexrelease>      \MessageBreak There is nothing here to make robust}%
279 <latexrelease>    \@eha
280 <latexrelease>  }%
281 <latexrelease>  {%
282 <latexrelease>    \ifundefined{\expandafter\@gobble\string#1\space}%
283 <latexrelease>    {%

```

```

284 \latexrelease> \expandafter\let\csname
285 \latexrelease> \expandafter\@gobble\string#1\space\endcsname=#1%
286 \latexrelease> \edef\reserved@a{\string#1}%
287 \latexrelease> \def\reserved@b{#1}%
288 \latexrelease> \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
289 \latexrelease> \edef#1{%
290 \latexrelease> \ifx\reserved@a\reserved@b
291 \latexrelease> \noexpand\x@protect\noexpand#1%
292 \latexrelease> \fi
293 \latexrelease> \noexpand\protect\expandafter\noexpand
294 \latexrelease> \csname\expandafter\@gobble\string#1\space\endcsname}%
295 \latexrelease> }%
296 \latexrelease> {\@latex@info{The control sequence ‘\string#1’ is already robust}}%
297 \latexrelease> }%
298 \latexrelease>}%
299 \latexrelease>\EndIncludeInRelease
300 \latexrelease>\IncludeInRelease{0000/00/00}{\MakeRobust}{\MakeRobust}%
301 \latexrelease>\let\MakeRobust\@undefined
302 \latexrelease>\EndIncludeInRelease
303 \kernel)

```

`\kernel@make@fragile` The opposite of `\MakeRobust` except that it doesn't do many checks as it is internal to the kernel. Why does one want such a thing? Only for compatibility reasons if `latexrelease` requests a rollback of the kernel. For this reason we pretend that this command existed in all earlier versions of L^AT_EX i.e., we are not rolling it back since we need it precisely then. But we have to get it into the `latexrelease` file so that a roll forward is possible too.

```

304 \kernel)
305 \kernel | latexrelease>
306 \latexrelease> \IncludeInRelease{0000/00/00}%
307 \latexrelease> \kernel@make@fragile{Undo robustness}%
308 \def\kernel@make@fragile#1{%
309 \@ifundefined{\expandafter\@gobble\string#1\space}%

```

If not robust do nothing.

```

310 {}%

```

Otherwise copy `\foo` back to `\foo` and then undefine the payload command.

```

311 {}%
312 \global\expandafter\let\expandafter #1\csname
313 \expandafter\@gobble\string#1\space\endcsname
314 \global\expandafter\let\csname
315 \expandafter\@gobble\string#1\space\endcsname\@undefined
316 }%
317 }
318 \latexrelease>\EndIncludeInRelease
319 \kernel | latexrelease>
320 \kernel)

```

11.5 Internal defining commands

These commands are used internally to define other L^AT_EX commands.

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

```

321 </2ekernel>
322 <latexrelease>\IncludeInRelease{2018-04-01}{\@ifundefined}
323 <latexrelease>\Leave commands undefined in \@ifundefined}%
324 <*2ekernel|latexrelease>

Version using \ifcsname to avoid defining undefined tokens to \relax. Defined
here to simplify using unmatched \fi.

325 \def\@ifundefined#1{%
326   \ifcsname#1\endcsname\@ifundefin@d@i\else\@ifundefin@d@ii\fi{#1}}

327 \long\def\@ifundefin@d@i#1\fi#2{\fi
328   \expandafter\ifx\csname #2\endcsname\relax
329     \@ifundefin@d@ii
330   \fi
331   \@secondoftwo}

332 \long\def\@ifundefin@d@ii\fi#1#2#3{\fi #2}

Now test of engine.

333 \ifx\numexpr\@undefined

Classic version (should not be needed as etex is assumed).

334 \def\@ifundefined#1{%
335   \expandafter\ifx\csname#1\endcsname\relax
336     \expandafter\@firstoftwo
337   \else
338     \expandafter\@secondoftwo
339   \fi}
340 \else\ifx\directlua\@undefined

Use the \ifcsname defined above.

341 \else

Optimised version for LuaTeX, using \lastnamedcs

342 \def\@ifundefined#1{%
343   \ifcsname#1\endcsname
344     \expandafter\ifx\lastnamedcs\relax\else\@ifundefin@d@i\fi
345   \fi
346   \@firstoftwo}

347 \long\def\@ifundefin@d@i#1#2#3#4#5{#1#2#5}

348 \fi
349 \fi
350 </2ekernel|latexrelease>
351 <latexrelease>\EndIncludeInRelease
352 <latexrelease>\IncludeInRelease{0000-00-00}{\@ifundefined}
353 <latexrelease>\Leave commands undefined in \@ifundefined}%
354 <latexrelease>\def\@ifundefined#1{%
355 <latexrelease>   \expandafter\ifx\csname#1\endcsname\relax
356 <latexrelease>     \expandafter\@firstoftwo
357 <latexrelease>   \else
358 <latexrelease>     \expandafter\@secondoftwo
359 <latexrelease>   \fi}
360 <latexrelease>\EndIncludeInRelease
361 <*2ekernel>

```

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

```
362 \edef\@qend{\expandafter\@cdr\string\end\@nil}
363 \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.

```
364 \long\def\@ifnextchar#1#2#3{%
365   \let\reserved@d=#1%
366   \def\reserved@a{#2}%
367   \def\reserved@b{#3}%
368   \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.

```
369 \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`.

```
370 \def\@ifnch{%
371   \ifx\@let@token\@sptoken
372     \let\reserved@c\@xifnch
373   \else
374     \ifx\@let@token\reserved@d
375       \let\reserved@c\reserved@a
376     \else
377       \let\reserved@c\reserved@b
378     \fi
379   \fi
380   \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.

```
381 \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.

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

`\@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`.

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

```

\@dblarg
\@xdblarg 384 \long\def\@dblarg#1{\kernel@ifnextchar[{#1}]{\@xdblarg{#1}}}
385 \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.

386 \def\@sanitize{\@makeother\ \@makeother\\\@makeother$\@makeother\&%
387 \@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.

388 \def \@onelevel@sanitize #1{%
389 \edef #1{\expandafter\strip@prefix
390 \meaning #1}%
391 }

\makeatletter Make internal control sequences accessible or inaccessible.
\makeatother 392 \DeclareRobustCommand\makeatletter{\catcode'\@11\relax}
393 \DeclareRobustCommand\makeatother{\catcode'\@12\relax}

```

12 Discretionary Hyphenation

```

\@dischph
\@dischph Moved here to be after the definition of \DeclareRobustCommand.
The primitive \@dischph command adds a discretionary hyphen using the current font’s
\hyphenchar. Monospace fonts are usually declared with \hyphenchar set to −1
to suppress hyphenation.
LATEX, from LATEX 2.09 in 1986 defined \@dischph by

\def\@dischph{-\discretionary{-}{-}{-}}

```

The following comment was added when these commands were first set up, 19 April 1986:

the \@dischph 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_EX definition is saved as \@@hyph just in case anyone needs it.

L^AT_EX 2_ε, between 1993 and 2017, had a comment at this point saying that the definition “would probably change” because the definition always uses −. The definition used below was given in comments at this point during time.

In 2017 we finally enabled this definition by default, with the older L^AT_EX definition accessible via `latexrelease` as usual.

```

394 \if2kernel)
395 (latexrelease)\IncludeInRelease{2017/04/15}{\@dischph}{Use \hyphenchar in \@dischph}%

```

Temporary definition of \@latex@info, final definition is later.

```

396 (*2ekernel)
397 \def\@latex@info#1{}
398 (/2ekernel)

399 (*2ekernel | latexrelease)
400 \DeclareRobustCommand{\-}{%
401   \discretionary{%
402     \char \ifnum\hyphenchar\font<\z@
403       \defaultshyphenchar
404     \else
405       \hyphenchar\font
406     \fi
407     }{}{}%
408 }
409 \let\@dischyph=\-
410 (/2ekernel | latexrelease)
411 (latexrelease)\EndIncludeInRelease
412 (latexrelease)\IncludeInRelease{0000/00/00}{\-}{Use \hyphenchar in \-}%
413 (latexrelease)\def\-\{\discretionary{-}{}{}\}
414 (latexrelease)\let\@dischyph=\-
415 (latexrelease)\EndIncludeInRelease
416 (*2ekernel)

    Delayed from ltvers.dtx
417 \newif\if@includeinrelease
418 \@includeinreleasefalse

    Delayed from ltplain.dtx
419 (/2ekernel)
420 (*2ekernel | latexrelease)
421 (latexrelease)\IncludeInRelease{2019/10/01}%
422 (latexrelease)          {\allowbreak}{Make various commands robust}%
423 \MakeRobust\allowbreak
424 \MakeRobust\bigbreak
425 \MakeRobust\break
426 \MakeRobust\dotfill
427 \MakeRobust\frenchspacing
428 \MakeRobust\goodbreak
429 \MakeRobust\hrulefill
430 \MakeRobust\medbreak
431 \MakeRobust\nobreak
432 \MakeRobust\nonfrenchspacing
433 \MakeRobust\obeylines
434 \MakeRobust\obeyspaces
435 \MakeRobust\slash
436 \MakeRobust\smallbreak
437 \MakeRobust\strut
438 \MakeRobust\underbar
439 (/2ekernel | latexrelease)
440 (latexrelease)\EndIncludeInRelease
441 (latexrelease)\IncludeInRelease{0000/00/00}%
442 (latexrelease)          {\allowbreak}{Make various commands robust}%
443 (latexrelease)
444 (latexrelease)\kernel@make@fragile\allowbreak

```

```

445 \latexrelease\kernel@make@fragile\bigbreak
446 \latexrelease\kernel@make@fragile\break
447 \latexrelease\kernel@make@fragile\dotfill
448 \latexrelease\kernel@make@fragile\frenchspacing
449 \latexrelease\kernel@make@fragile\goodbreak
450 \latexrelease\kernel@make@fragile\hrulefill
451 \latexrelease\kernel@make@fragile\medbreak
452 \latexrelease\kernel@make@fragile\nobreak
453 \latexrelease\kernel@make@fragile\nonfrenchspacing
454 \latexrelease\kernel@make@fragile\obeylines
455 \latexrelease\kernel@make@fragile\obeyspaces
456 \latexrelease\kernel@make@fragile\slash
457 \latexrelease\kernel@make@fragile\smallbreak
458 \latexrelease\kernel@make@fragile\strut
459 \latexrelease\kernel@make@fragile\underbar
460 \latexrelease\
461 \latexrelease\EndIncludeInRelease
462 \*2ekernel\
463 \2ekernel\

```


File e

lalloc.dtx

13 Counters

This section deals with counter and other variable allocation.

1 (*2ekernel)

The following are from plain T_EX:

\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 10001–10004.

\@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^AT_EX kernel commands.

\@tempcntb 7 \newcount\@tempcnta

8 \newcount\@tempcntb

\if@tempswa General boolean switch used by L^AT_EX kernel commands.

9 \newif\if@tempswa

\@tempdima Scratch dimen registers used by L^AT_EX kernel commands.

\@tempdimb 10 \newdimen\@tempdima

\@tempdimc 11 \newdimen\@tempdimb

12 \newdimen\@tempdimc

\@tempboxa Scratch box register used by L^AT_EX kernel commands.

13 \newbox\@tempboxa

\@tempskipa Scratch skip registers used by L^AT_EX 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

14 Program control structure

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

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```

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

```

End of historical L^AT_EX 2.09 comments.

```

\@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}

```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```

\@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}
End of historical LATEX 2.09 comments.

\@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
\@iffileonpath 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 </2kernel>

```

File g

lterror.dtx

15 Error handling and tracing

This section defines L^AT_EX'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
```

15.1 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_EX 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_EX'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_EX! 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_EX. 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_EX'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}%

```

<pre> \PackageError \PackageWarning \PackageWarningNoLine \PackageInfo \ClassError \ClassWarning \ClassWarningNoLine \ClassInfo </pre>	<pre> \PackageError{<package>}{<error>}{<help>} \PackageWarning{<package>}{<warning>} \PackageWarningNoLine{<package>}{<warning>} \PackageInfo{<package>}{<info>} </pre>
--	--

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’.
165 \def\on@line{ on input line \the\inputlineno}

```

`\@warning` Older L^AT_EX 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`

```
166 \let\@warning\@latex@warning
167 \let\@@warning\@latex@warning@no@line
168 \global\let\@latexerr\@latex@error
```

`\@spaces` Four spaces.

```
169 \def\@spaces{\space\space\space\space}
```

15.2 Specific errors

`\@eha` The more common error help messages.

```
\@ehb 170 \gdef\@eha{%
\@ehc 171 Your command was ignored.\MessageBreak
\@ehd 172 Type \space I <command> <return> \space to replace it %
173 with another command,\MessageBreak
174 or \space <return> \space to continue without it.}
175 \gdef\@ehb{%
176 You've lost some text. \space \@ehc}
177 \gdef\@ehc{%
178 Try typing \space <return> %
179 \space to proceed.\MessageBreak
180 If that doesn't work, type \space X <return> \space to quit.}
181 \gdef\@ehd{%
182 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...`

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

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

```
189 \gdef\@nolnerr{%
190 \@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^AT_EX 2.09 by `\setcounter`, `\addtocounter` or `\newcounter` for undefined counter. DO NOT use for L^AT_EX 2_ε it MIGHT vanish! Use `\@nocounterr{<cnt>}` instead.

```
191 \gdef\@nocounterr#1{%
192 \@latex@error{No counter '#1' defined}\@eha}
193 \gdef\@nocnterr{\@nocounterr?}
```

`\@ctrerr` Called when trying to print the value of a counter numbered by letters that's greater than 26.

```

194 \gdef\@ctrerr{%
195   \latex@error{Counter too large}\@ehb}

```

`\@nocument` Error produced if paragraphs are typeset in the preamble.

```

196 \gdef\@nocument{%
197   \latex@error{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.

```

198 \gdef\@badend#1{%
199   \latex@error{\protect\begin{\@currenvir}\@currenvline
200               \space ended by \protect\end{#1}}\@eha}

```

`\@badmath` Called by `\[, \], \(< or \)` when used in wrong mode.

```

201 \gdef\@badmath{%
202   \latex@error{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.

```

203 \gdef\@toodeep{%
204   \latex@error{Too deeply nested}\@ehd}

```

`\@badpoptabs` Called by `\endtabbing` when not enough `\poptabs` have occurred, or by `\poptabs` when too many have occurred.

```

205 \gdef\@badpoptabs{%
206   \latex@error{\protect\pushtabs\space and \protect\poptabs
207               \space don't match}\@ehd}

```

`\@badtab` Called by `\>, \+, \- or \<` when stepping to an undefined tab.

```

208 \gdef\@badtab{%
209   \latex@error{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`.

```

210 \gdef\@preamerr#1{%
211   \begingroup
212     \let\protect\relax
213     \latex@error{\ifcase #1 Illegal character\or
214                 Missing @-exp\or Missing p-arg\fi\space
215                 in array arg}\@ehd
216   \endgroup}

```

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

```

217 \gdef\@badlinearg{%
218   \latex@error{%
219       Bad \protect\line\space or \protect\vector
220       \space argument}\@ehb}

```

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

```

221 \gdef\@parmoderr{%
222   \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.

```

223 \gdef\@fltovf{%
224   \latex@error{Too many unprocessed floats}\@ehb}

```

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

```

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

```

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

```

227 %\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`.

```

228 \gdef\@noitemerr{%
229   \latex@error{Something's wrong--perhaps a missing %
230     \protect\item}\@ehc}

```

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

```

231 \gdef\@notprerr{%
232   \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.

```

233 \gdef\@inmatherr#1{%
234   \relax
235   \ifmmode
236     \latex@error{Command \protect#1 invalid in math mode}\@ehc
237   \fi}

```

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

```

238 %\def\@invalidchar{\latex@error{Invalid character in input}\@ehc}

```

As well as the above error commands some error messages are directly coded to save space. The messages already present in \LaTeX 2.09 include:

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.

15.3 Tracing

The `trace` package implements the commands `\traceon` and `\traceoff` that work similar to `\tracingall` but skip certain code blocks that produce a lot of tracing output being of no interest during debugging (for example loading a font). Code blocks that should be hidden during tracing need to be surrounded by the macros `\conditionally@traceoff` and `\conditionally@traceon`.

For the kernel code the `trace` package then redefines a number of macros to include this tracing support.

However, in order to allow any macro package to react to `\traceon` we also provide dummy definitions for the two commands in the kernel so that they can be used by external packages without the need to distinguish between `trace` being loaded or not.

```
\conditionally@traceon  These are only dummy definitions. For details see the trace package.
\conditionally@traceoff 239 \let\conditionally@traceon\@empty
                        240 \let\conditionally@traceoff\@empty
                        241 </2kernel>
```

File h

ltpar.dtx

16 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.

16.1 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}
```

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

```
5 \def\@restorepar{\def\par{\@par}}
6 \*2kernel)
```

File i

ltspace.dtx

17 Spacing

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

17.1 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.

17.2 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}
```

17.3 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_EX itself.

17.4 The code

```

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

3 (/2ekernel)
4 (*2ekernel | latexrelease)
5 (latexrelease)\IncludeInRelease{2019/10/01}%
6 (latexrelease)          {\pagebreak}{Make commands robust}%

\pagebreak
\nopagebreak 7 \DeclareRobustCommand\pagebreak{\@testopt{\@no@pgbk-}4}
8 \DeclareRobustCommand\nopagebreak{\@testopt{\@no@pgbk4}

\linebreak
\nolinebreak 9 \DeclareRobustCommand\linebreak{\@testopt{\@no@lnbk-}4}
10 \DeclareRobustCommand\nolinebreak{\@testopt{\@no@lnbk4}

\samepage
11 \DeclareRobustCommand\samepage{\interlinepenalty\@M
12   \postdisplaypenalty\@M
13   \interdisplaylinepenalty\@M
14   \@beginparpenalty\@M
15   \@endparpenalty\@M
16   \@itempenalty\@M
17   \@secpenalty\@M
18   \interfootnotelinepenalty\@M}

19 (/2ekernel | latexrelease)
20 (latexrelease)\EndIncludeInRelease
21 (latexrelease)\IncludeInRelease{0000/00/00}%
22 (latexrelease)          {\pagebreak}{Make commands robust}%
23 (latexrelease)

24 (latexrelease)\kernel@make@fragile\pagebreak
25 (latexrelease)\kernel@make@fragile\nopagebreak
26 (latexrelease)\kernel@make@fragile\linebreak
27 (latexrelease)\kernel@make@fragile\nolinebreak
28 (latexrelease)\kernel@make@fragile\samepage
29 (latexrelease)
30 (latexrelease)\EndIncludeInRelease
31 (*2ekernel)

```

`\@no@pgbk`

```
32 \def\@no@pgbk #1[#2]{%
33   \ifvmode
34     \penalty #1\@getpen{#2}%
35   \else
36     \@bsphack
37     \vadjust{\penalty #1\@getpen{#2}}%
38     \@esphack
39   \fi}
```

`\@no@lnbk`

```
40 \def\@no@lnbk #1[#2]{%
41   \ifvmode
42     \@nolnerr
43   \else
44     \@tempskipa\lastskip
45     \unskip
46     \penalty #1\@getpen{#2}%
47     \ifdim\@tempskipa>\z@
48       \hskip\@tempskipa
49       \ignorespaces
50   \fi
51   \fi}
```

`\@` 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 `\@`.

```
52 </2kernel>
53 <*2kernel | latexrelease>
54 <latexrelease> \IncludeInRelease{2020/02/02}%
55 <latexrelease> {\@normalcr}{Make robust}%
56 \protected\def\@normalcr{%
57   \let \reserved@e \relax
58   \let \reserved@f \relax
59   \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
60     \xnewline}%
61   \@xnewline}
```

```

62 \let\\\@normalcr
63 /2ekernel | latexrelease)
64 (latexrelease)\EndIncludeInRelease
65 (latexrelease)\IncludeInRelease{0000/00/00}%
66 (latexrelease)          {\@normalcr}{Make robust}%
67 (latexrelease)
68 (latexrelease)\DeclareRobustCommand\\\{%
69 (latexrelease)  \let \reserved@e \relax
70 (latexrelease)  \let \reserved@f \relax
71 (latexrelease)  \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
72 (latexrelease)          \@xnewline}%
73 (latexrelease)  \@xnewline}
74 (latexrelease)\expandafter\let\expandafter\@normalcr
75 (latexrelease)  \csname\expandafter\@gobble\string\\\ \endcsname
76 (latexrelease)
77 (latexrelease)\EndIncludeInRelease
78 (*2ekernel)

```

`\newline` A simple form of the ‘normal’ definition of `\\`.

```
79 \DeclareRobustCommand\newline{\@normalcr\relax}
```

`\@xnewline`

```

80 \def\@xnewline{\@ifnextchar[% ] bracket matching
81             \@newline
82             {\@gnewline\relax}}

```

`\@newline`

```

83 \def\@newline[#1]{\let \reserved@e \vadjust
84             \@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

```

85 \def\@gnewline #1{%
86   \ifvmode
87     \@nolnerr
88   \else
89     \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break
90   \fi}

```

`\@getpen`

```

91 \def\@getpen#1{\ifcase #1 \z@ \or \@lowpenalty\or
92             \@medpenalty \or \@highpenalty
93             \else \@M \fi}

```

`\if@nobreak` Switch used to avoid page breaks caused by `\label` after a section heading, etc. It should be **GLOBALLY** set true after the `\nobreak` and **globally** set false by the next invocation of `\everypar`.

Commands that reset `\everypar` should globally set it false if appropriate.

```

94 \def\@nobreakfalse{\global\let\if@nobreak\iffalse}
95 \def\@nobreaktrue {\global\let\if@nobreak\iftrue}
96 \@nobreakfalse

```

`\@savsk` Registers used to save the space factor and last skip.
`\@savsf` 97 `\newdimen\@savsk`
98 `\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 environments (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
      \fi
    \fi
}
```



```

        \ifdim \@savsk>\z@
        \ignorespaces
        \fi
    \fi
\fi
}

```

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

```

99 \def\@bsphack{%
100 \relax
101 \ifhmode
102 \savsk\lastskip
103 \savsf\spacefactor
104 \fi}

```

\@esphack Companion to \@bsphack. If this command is not properly paired with \@bsphack one might end up with a low-level T_EX error: “BAD spacefactor”. One possible cause is calling \@bsphack in vertical mode, then doing something that gets you (sometimes) into horizontal mode and finally calling \@esphack. Even if no error is generated that is wrong, because \@esphack will then use the saved values for \@savsk and \@savsf from some earlier invocation of \@bsphack which will have nothing to do with the current situation.

```

105 </2ekernel>
106 <latexrelease>\IncludeInRelease{2018/10/10}%
107 <latexrelease>          {\@esphack}{hyphenation and nobreak after space hack}%
108 <*2ekernel | latexrelease>
109 \def\@esphack{%
110 \relax
111 \ifhmode
112 \spacefactor\@savsf
113 \ifdim\@savsk>\z@
114 \ifdim\lastskip=\z@
115 \nobreak \hskip\z@skip
116 \fi
117 \ignorespaces
118 \fi
119 \else
120 \ifvmode
121 \if@nobreak\nobreak\else\if@noskipsec\nobreak\fi\fi
122 \fi
123 \fi}%
124 </2ekernel | latexrelease>
125 <latexrelease>\EndIncludeInRelease
126 <latexrelease>\IncludeInRelease{2015/10/01}%
127 <latexrelease>          {\@esphack}{hyphenation and nobreak after space hack}%
128 <latexrelease>\def\@esphack{%
129 <latexrelease> \relax
130 <latexrelease> \ifhmode
131 <latexrelease> \spacefactor\@savsf
132 <latexrelease> \ifdim\@savsk>\z@

```

```

133 <latexrelease> \ifdim\lastskip=\z@
134 <latexrelease> \nobreak \hskip\z@skip
135 <latexrelease> \fi
136 <latexrelease> \ignorespaces
137 <latexrelease> \fi
138 <latexrelease> \fi}%
139 <latexrelease>\EndIncludeInRelease
140 <latexrelease>\IncludeInRelease{2015/01/01}%
141 <latexrelease> {\@esphack}{hyphenation and nobreak after space hack}%
142 <latexrelease>\def\@esphack{%
143 <latexrelease> \relax
144 <latexrelease> \ifhmode
145 <latexrelease> \spacefactor\@savsf
146 <latexrelease> \ifdim\@savsk>\z@
147 <latexrelease> \nobreak \hskip\z@skip
148 <latexrelease> \ignorespaces
149 <latexrelease> \fi
150 <latexrelease> \fi}%
151 <latexrelease>\EndIncludeInRelease
152 <latexrelease>\IncludeInRelease{0000/00/00}%
153 <latexrelease> {\@esphack}{hyphenation and nobreak after space hack}%
154 <latexrelease>\def\@esphack{%
155 <latexrelease> \relax
156 <latexrelease> \ifhmode
157 <latexrelease> \spacefactor\@savsf
158 <latexrelease> \ifdim\@savsk>\z@
159 <latexrelease> \ignorespaces
160 <latexrelease> \fi
161 <latexrelease> \fi}%
162 <latexrelease>\EndIncludeInRelease
163 <*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
164 </2ekernel>
165 <latexrelease>\IncludeInRelease{2015/01/01}%
166 <latexrelease> {\@Esphack}{hyphenation after space hack}%
167 <*2ekernel | latexrelease>
168 \def\@Esphack{%
169 \relax
170 \ifhmode
171 \spacefactor\@savsf
172 \ifdim\@savsk>\z@
173 \nobreak \hskip\z@skip
174 \@ignoretrue
175 \ignorespaces
176 \fi
177 \fi}%
178 </2ekernel | latexrelease>
179 <latexrelease>\EndIncludeInRelease
180 <latexrelease>\IncludeInRelease{0000/00/00}%
181 <latexrelease> {\@Esphack}{hyphenation after space hack}%
182 <latexrelease>\def\@Esphack{%

```

```

183 <latexrelease> \relax
184 <latexrelease> \ifhmode
185 <latexrelease> \spacefactor\@savs f
186 <latexrelease> \ifdim\@savsk>\z@
187 <latexrelease> \ignoretrue
188 <latexrelease> \ignorespaces
189 <latexrelease> \fi
190 <latexrelease> \fi}%
191 <latexrelease>\EndIncludeInRelease
192 <*2kernel>

```

`\@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
    \@savs f \spacefactor
  \else
    \ifhmode
      \@savsk \lastskip
      \@savs f \spacefactor
    \fi
  \fi
}

```

17.5 Vertical spacing

L^AT_EX supports the plain T_EX 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.

```

193 \def\@xaddvskip{%
194   \ifdim\lastskip<\@tempskipb
195     \vskip-\lastskip
196     \vskip\@tempskipb
197   \else
198     \ifdim\@tempskipb<\z@
199       \ifdim\lastskip<\z@
200         \else
201           \advance\@tempskipb\lastskip
202           \vskip-\lastskip
203           \vskip \@tempskipb
204         \fi
205       \fi
206     \fi}

```

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

```

207 \def\addvspace#1{%
208   \ifvmode
209     \if@minipage\else
210       \ifdim \lastskip =\z@
211         \vskip #1\relax
212       \else
213         \@tempskipb#1\relax
214         \@xaddvskip
215       \fi
216     \fi
217   \else
218     \@noitemerr
219   \fi}

```

\addpenalty

```

220 \if2ekernel
221 \iflatexrelease\IncludeInRelease{2015/01/01}%
222 \iflatexrelease
223 \if*2ekernel\iflatexrelease

```

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.

```
224 \def\addpenalty#1{%
```

```
225   \ifvmode
```

```
226     \if@minipage
```

```
227     \else
```

```
228       \if@nobreak
```

```
229       \else
```

```
230         \ifdim\lastskip=\z@
```

```
231         \penalty#1\relax
```

```
232       \else
```

```
233         \@tempskipb\lastskip
```

We have to make sure the final `\vskip` seen by \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 \TeX .

```
234     \begingroup
```

```
235       \@tempskipa\@tempskipb
```

```
236       \advance \@tempskipb
```

```
237       \ifdim\prevdepth>\maxdepth\maxdepth\else
```

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

```
238         \ifdim \prevdepth = -\@m\p@ \z@ \else \prevdepth \fi
```

```
239       \fi
```

```
240       \vskip -\@tempskipb
```

```
241       \penalty#1%
```

```
242       \ifdim\@tempskipa=\@tempskipb
```

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

```
243     \else
```

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

```
244       \advance\@tempskipb -\@tempskipa
```

```
245       \vskip \@tempskipb
```

```
246     \fi
```

The final skip is always the specified length.

```
247       \vskip \@tempskipa
```

```
248     \endgroup
```

```
249   \fi
```

```
250 \fi
```

```
251 \fi
```

```
252 \else
```

```
253   \@noitemerr
```

```
254 \fi}%
```

```
255 </2ekernel | latexrelease>
```

```
256 <latexrelease>\EndIncludeInRelease
```

```
257 <latexrelease>\IncludeInRelease{0000/00/00}%
```

```
258 <latexrelease>      {\addpenalty}{\addpenalty}%
```

```
259 <latexrelease>\def\addpenalty#1{%
```

```
260 <latexrelease>  \ifvmode
```

```
261 <latexrelease>    \if@minipage
```

```
262 <latexrelease>    \else
```

```
263 <latexrelease>      \if@nobreak
```

```
264 <latexrelease>      \else
```

```
265 <latexrelease>        \ifdim\lastskip=\z@
```

```
266 <latexrelease>        \penalty#1\relax
```

```

267 \latexrelease\else
268 \latexrelease\@tempskipb\lastskip
269 \latexrelease\vskip -\lastskip
270 \latexrelease\penalty#1%
271 \latexrelease\vskip\@tempskipb
272 \latexrelease\fi
273 \latexrelease\fi
274 \latexrelease\fi
275 \latexrelease\else
276 \latexrelease\@noitemerr
277 \latexrelease\fi}%
278 \latexrelease\EndIncludeInRelease
279 \*2ekernel)

```

`\vspace` The new code for these commands depends on the following facts:
`\@vspace`
`\@vspacer`

- 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.

```

280 \DeclareRobustCommand\vspace{\@ifstar\@vspacer\@vspace}
281 \def\@vspace #1{%
282   \ifvmode
283     \vskip #1
284     \vskip\z@skip
285   \else
286     \@bsphack
287     \vadjust{\@restorepar
288               \vskip #1
289               \vskip\z@skip
290             }%
291     \@esphack
292   \fi}
293 \def\@vspacer#1{%
294   \ifvmode
295     \dimen@\prevdepth
296     \hrule \@height\z@
297     \nobreak
298     \vskip #1
299     \vskip\z@skip
300     \prevdepth\dimen@
301   \else
302     \@bsphack
303     \vadjust{\@restorepar
304               \hrule \@height\z@
305               \nobreak
306               \vskip #1
307               \vskip\z@skip}%
308     \@esphack
309   \fi}

```

```

\smallskip
\medskip 310 \def\smallskip{\vspace\smallskipamount}
\bigskip 311 \def\medskip{\vspace\medskipamount}
          312 \def\bigskip{\vspace\bigskipamount}

\smallskipamount
\medskipamount 313 \newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
\bigskipamount 314 \newskip\medskipamount \medskipamount =6pt plus 2pt minus 2pt
          315 \newskip\bigskipamount \bigskipamount =12pt plus 4pt minus 4pt

```

17.6 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!

```

316 \DeclareRobustCommand{\nobreakdashes}{%
317   \leavevmode
318   \toks@{}%
319   \def\reserved@a##1{\toks@\expandafter{\the\toks@-}%
320     \futurelet\@let@token \reserved@b}%
321   \def\reserved@b  {\ifx\@let@token -%
322     \expandafter\reserved@a
323     \else
324     \setbox\z@ \hbox{\the\toks@\nobreak}%
325     \unhbox\z@
326     \spacefactor\sfcode'\-
327     \fi}%
328   \futurelet\@let@token \reserved@b
329 }

```

`\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 \LaTeX 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`.

```

330 \DeclareRobustCommand{\nobreakspace}{%
331   \leavevmode\nobreak\ }
332 \catcode '\~ =13
333 \def~{\nobreakspace}
334 \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.’’

```
335 \DeclareRobustCommand{\,}{%
336   \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi
337 }
```

`\@` 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.

```
338 \/2ekernel)
339 \latexrelease\IncludeInRelease{2015/01/01}%
340 \latexrelease          {\@}{Space after \@}%
341 \*2ekernel | latexrelease)
342 \def\@{\spacefactor\@m{}}%
343 \/2ekernel | latexrelease)
344 \latexrelease\EndIncludeInRelease
345 \latexrelease\IncludeInRelease{0000/00/00}%
346 \latexrelease          {\@}{Space after \@}%
347 \latexrelease\def\@{\spacefactor\@m}%
348 \latexrelease\EndIncludeInRelease
349 \*2ekernel)
```

`\hspace`

```
350 \DeclareRobustCommand\hspace{\@ifstar\@hspacer\@hspace}
```

`\@hspace`

```
351 \def\@hspace#1{\hskip #1\relax}
```

`\@hspacer`

extra `\hskip` Opt added 1985/17/12 to guard against a following `\unskip \relax` added 13 Oct 88 for usual T_EX lossage replaced both changes by `\hskip\z@skip` 27 Nov 91

```
352 \def\@hspacer#1{\vrule \@width\z@\nobreak
353   \hskip #1\hskip \z@skip}
```

`\fill`

```
354 \newskip\fill
355 \fill = Opt plus 1fill
```

`\stretch`

```
356 \def\stretch#1{\z@ \@plus #1fill\relax}
357 \/2ekernel)
358 \*2ekernel | latexrelease)
359 \latexrelease\IncludeInRelease{2018/12/01}%
360 \latexrelease          {\thinspace}{Start LR-mode}%
```

`\thinspace`

`\negthinspace`

`\enspace`

```
361 \DeclareRobustCommand\thinspace{\leavevmode@ifvmode\kern .16667em }
362 \DeclareRobustCommand\negthinspace{\leavevmode@ifvmode\kern-.16667em }
363 \DeclareRobustCommand\enspace{\leavevmode@ifvmode\kern.5em }
```

`\leavevmode@ifvmode`

Leave vmode but only if we are really in vmode, otherwise the expansion is empty (which is not the case with the default definition).

```
364 \protected\def\leavevmode@ifvmode{\ifvmode\expandafter\indent\fi}
```



```

365 </2ekernel | latexrelease>
366 <latexrelease>\EndIncludeInRelease
367 <latexrelease>\IncludeInRelease{0000/00/00}%
368 <latexrelease>{\thinspace}{Start LR-mode}%
369 <latexrelease>\def\thinspace{\kern .16667em }
370 <latexrelease>\def\negthinspace{\kern-.16667em }
371 <latexrelease>\def\enspace{\kern.5em }
372 <latexrelease>\let\leavevmode@ifvmode\@undefined
373 <latexrelease>\EndIncludeInRelease
374 <*2ekernel>

\enskip
\quad 375 \def\enskip{\hskip.5em\relax}
\qqquad 376 \def\quad{\hskip1em\relax}
377 \def\qqquad{\hskip2em\relax}

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

378 {\catcode'\^^M=13 \gdef\obeycr{\catcode'\^^M13 \def^^M{\\}\relax}%
379 \gobblecr}%
380 {\catcode'\^^M=13 \gdef\gobblecr{\@ifnextchar
381 \@gobble\ignorespaces}}
382 \gdef\restorecr{\catcode'\^^M5 }}

383 </2ekernel>

```

File j

ltlogos.dtx

18 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 <*2ekernel>
2 \DeclareRobustCommand\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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\varepsilon}$ 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 </2ekernel>
```

File k

ltxfiles.dtx

19 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}{\<then>}{\<else>}</code> If the file exists on the system, execute <i>then</i> otherwise execute <i>else</i> .
<code>\InputIfFileExists</code>	<code>{\NAME}{\<then>}{\<else>}</code> If the file exists on the system, execute <i>then</i> and input NAME otherwise execute <i>else</i> . <i>Historical \LaTeX 2.09 comments (not necessarily accurate any more):</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>\@writeckpt</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

End of historical LATEX 2.09 comments.

\@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
  11 \</2ekernel>
  12 \<latexrelease>\IncludeInRelease{2017/04/15}%
  13 \<latexrelease> {\document}{Save language for hyphenation}%
  14 \<*2ekernel|latexrelease>

Cancel the \begingroup from \begin.
  15 \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.
  16 \ifx\@unusedoptionlist\@empty\else
  17 \latex@warning@no@line{Unused global option(s):^^J%
  18 \spaces[\@unusedoptionlist]}%
  19 \fi
  20 \@colht\textheight
  21 \@colroom\textheight \vsize\textheight
  22 \columnwidth\textwidth
  23 \@clubpenalty\clubpenalty
  24 \if@twocolumn
  25 \advance\columnwidth -\columnsep
  26 \divide\columnwidth\tw@ \hsize\columnwidth \@firstcolumntrue
  27 \fi
  28 \hsize\columnwidth \linewidth\hsize
  29 \begingroup\@floatplacement\@dblfloatplacement
  30 \makeatletter\let\@writefile\@gobbletwo

```

```

31   \global \let \@multiplelabels \relax
32   \@input{\jobname.aux}%
33 \endgroup
34 \if@filesw
35   \immediate\openout\@mainaux\jobname.aux
36   \immediate\write\@mainaux{\relax}%
37 \fi

```

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

```

38   \process@table
39   \let\glb@currsizel@empty % Force math initialization.

40   \normalsize
41   \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 `\normalsfcodes` directly, in case that definition will be used. (This is an old bug, problem existed in L^AT_EX2.0x and plain T_EX.)

```

42   \ifx\normalsfcodes\empty
43     \ifnum\sfcodes'\.=\@m
44       \let\normalsfcodes\frenchspacing
45     \else
46       \let\normalsfcodes\nonfrenchspacing
47   \fi
48 \fi

```

For similar reasons also save the default language, this will be reset locally in the output routine. In particular it allows hyphenation in the page head even if the page break happens in verbatim. If this has already been set by a package, set to the value of `\language` at this point.

```

49   \ifx\document@default@language\m@ne
50     \chardef\document@default@language\language
51   \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.

```

52   \@noskipsecfalse

53   \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...

```

54   \let\AtBeginDocument\@firstofone
55   \@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.

```

56 \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
57 \global\@maxdepth\maxdepth
58 \global\let\@begindocumenthook\@undefined
59 \ifx\@listfiles\@undefined
60   \global\let\@filelist\relax
61   \global\let\@addtofilelist\@gobble
62 \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.

```

63 \gdef\do##1{\global\let ##1\@notprerr}%
64 \@preamblecmds

```

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

```

65 \global\let \@nodocument \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.

```

66 \global\let\do\noexpand

```

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

```

67 \ignorespaces}
68 \</2ekernel|latexrelease>
69 \<latexrelease>\EndIncludeInRelease
70 \<latexrelease>\IncludeInRelease{0000/00/00}%
71 \<latexrelease> {\document}{Save language for hyphenation}
72 \<latexrelease>\def\document{\endgroup
73 \<latexrelease> \ifx\@unusedoptionlist\@empty\else
74 \<latexrelease>   \<latex@warning@no@line>{Unused global option(s):^^J%
75 \<latexrelease>     \<spaces>[\@unusedoptionlist]}}%
76 \<latexrelease> \fi
77 \<latexrelease> \<colht>\textheight
78 \<latexrelease> \<colroom>\textheight \<vsize>\textheight
79 \<latexrelease> \<columnwidth>\textwidth
80 \<latexrelease> \<clubpenalty>\clubpenalty
81 \<latexrelease> \if@twocolumn
82 \<latexrelease>   \<advance>\columnwidth -\columnsep
83 \<latexrelease>   \<divide>\columnwidth\tw@ \<hsize>\columnwidth
84 \<latexrelease>   \<@firstcolumntrue>
85 \<latexrelease> \fi
86 \<latexrelease> \<hsize>\columnwidth \<linewidth>\hsize
87 \<latexrelease> \<begingroup>\<floatplacement>\<dblfloatplacement>
88 \<latexrelease>   \<makeatletter>\let\<writefile>\<gobbletwo>
89 \<latexrelease>   \global \let \<multiplelabels> \relax
90 \<latexrelease>   \<input>\jobname.aux}%
91 \<latexrelease> \endgroup
92 \<latexrelease> \if@filesw
93 \<latexrelease>   \<immediate>\openout\<mainaux>\jobname.aux
94 \<latexrelease>   \<immediate>\write\<mainaux>\relax}%
95 \<latexrelease> \fi

```

```

96 \latexrelease \process@table
97 \latexrelease \let\glb@currsizel@empty
98 \latexrelease \normalsize
99 \latexrelease \everypar{}%
100 \latexrelease \ifx\normalsfcode@empty
101 \latexrelease \ifnum\scode'\.=\@m
102 \latexrelease \let\normalsfcode\xfrenchspacing
103 \latexrelease \else
104 \latexrelease \let\normalsfcode\nonfrenchspacing
105 \latexrelease \fi
106 \latexrelease \fi
107 \latexrelease \@noskipsecfalse
108 \latexrelease \let \@refundefined \relax
109 \latexrelease \let\AtBeginDocument\@firstofone
110 \latexrelease \@begindocumenthook
111 \latexrelease \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
112 \latexrelease \global\@maxdepth\maxdepth
113 \latexrelease \global\let\@begindocumenthook\@undefined
114 \latexrelease \ifx\@listfiles\@undefined
115 \latexrelease \global\let\@filelist\relax
116 \latexrelease \global\let\@addtofilelist\@gobble
117 \latexrelease \fi
118 \latexrelease \gdef\do##1{\global\let ##1\@notprerr}%
119 \latexrelease \@preamblecmds
120 \latexrelease \global\let \@nodocument \relax
121 \latexrelease \global\let\do\noexpand
122 \latexrelease \ignorespaces}
123 \latexrelease\EndIncludeInRelease
124 (*2ekernel)

125 \@onlypreamble\document

```

\normalsfcode 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.

```
126 \let\normalsfcode@empty
```

\nofiles Set \@fileswfalse which suppresses the places where L^AT_EX 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.

```

127 \def\nofiles{%
128   \@fileswfalse
129   \typeout{No auxiliary output files.^^J}%
130   \long\def\protected@write##1##2##3%
131     {\write\m@ne{}\if@nobreak\ifvmode\nobreak\fi\fi}%
132   \let\makeindex\relax
133   \let\makeglossary\relax}
134 \@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`.

```

135 \long\def \protected@write#1#2#3{%
136     \begingroup
137     \let\thepage\relax
138     #2%
139     \let\protect\@unexpandable@protect
140     \edef\reserved@a{\write#1{#3}}%
141     \reserved@a
142     \endgroup
143     \if@nobreak\ifvmode\nobreak\fi\fi
144 }

```

```

145 \let\@auxout=\@mainaux

```

`\includeonly`

```

146 </2kernel>
147 <*2kernel | latexrelease>
148 <latexrelease>\IncludeInRelease{2019/10/01}%
149 <latexrelease>           {\includeonly}{Spaces in file names}%
150 \def\includeonly#1{%
151     \@partswtrue
152     \set@curr@file{\zap@space#1 \@empty}%
153     \let\@partlist\@curr@file
154 }
155 \@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 L^AT_EX 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.

```

156 \def\include#1{\relax
157     \ifnum\@auxout=\@partaux
158         \@latex@error{\string\include\space cannot be nested}\@eha
159     \else
160         \set@curr@file{#1 }%
161         \expandafter\@include\@curr@file
162     \fi}

163 </2kernel | latexrelease>
164 <latexrelease>\EndIncludeInRelease
165 <latexrelease>\IncludeInRelease{0000/00/00}%
166 <latexrelease>           {\includeonly}{Spaces in file names}%
167 <latexrelease>\def\includeonly#1{%
168 <latexrelease>     \@partswtrue
169 <latexrelease>     \edef\@partlist{\zap@space#1 \@empty}}
170 <latexrelease>
171 <latexrelease>\def\include#1{\relax
172 <latexrelease>     \ifnum\@auxout=\@partaux

```

```

173 \latexrelease) \latexerror{\string\include\space cannot be nested}\@eha
174 \latexrelease) \else \@include#1 \fi}
175 \latexrelease)
176 \latexrelease)\EndIncludeInRelease
177 (*2ekernel)

\@include
178 \def\@include#1 {%
179 \clearpage
180 \if@filesw
181 \immediate\write\@mainaux{\string\@input{#1.aux}}}%
182 \fi
183 \@tempswatrue
184 \if@partsw
185 \@tempswafalse
186 \edef\reserved@a{\@partlist\do
187 \ifx\reserved@a\reserved@b\@tempswatrue\fi}%
188 \fi
189 \if@tempswa
190 \let\@auxout\@partaux
191 \if@filesw
192 \immediate\openout\@partaux #1.aux
193 \immediate\write\@partaux{\relax}%
194 \fi
195 \input@{#1.tex}%
196 \clearpage
197 \@writeckpt{#1}%
198 \if@filesw
199 \immediate\closeout\@partaux
200 \fi
201 \else
202 If the file is not included, reset \deadcycles, so that a long list of non-included
203 files does not generate an ‘Output loop’ error.
204 \deadcycles\z@
205 \nameuse{cp@#1}%
206 \fi
207 \let\@auxout\@mainaux}

\@writeckpt
207 \def\@writeckpt#1{%
208 \if@filesw
209 \immediate\write\@partaux{\string\@setckpt{#1}\@charlb}%
210 {\let\@elt\@wckptelt \cl@ckpt}%
211 \immediate\write\@partaux{\@charrb}%
212 \fi}

\@wckptelt
213 \def\@wckptelt#1{%
214 \immediate\write\@partaux{%
215 \string\setcounter{#1}{\the\@nameuse{c@#1}}}}

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

```

```

\@charlb The following defines \@charlb and \@charrb to be { and }, respectively with
\@charrb \catcode 11.
217 {\catcode' [=1 \catcode']=2
218 \catcode' {=11 \catcode'}=11
219 \gdef\@charlb[{
220 \gdef\@charrb[]
221 ]% }brace matching

```

19.1 Safe Input Macros

`\@curr@file` File name handling is done by generating a csname from the provided file name
`\set@curr@file` (which means that UTF-8 octets gets turned into strings as this is what happens if they appear in a csname due to the code in `utf8.def`). By setting `\escapechar` to -1 we ensure that we don't get a backslash in front. As a result we end up with all characters as catcode 12 (plus spaces). We then sometimes add quotes around the construct (removing any existing inner quotes. Sometimes we only remove the quotes if they have been supplied by the user. There is clearly some room for improvement.

A side effect of the new code is that we will see quotes around file name displays where there haven't been any before.

For compatibility with existing code using `{abc}.tex` or `{one.two}.png` an initial brace group is discarded before expansion and `\string` is applied. The content of the brace group is discarded. This means that a leading space will be lost unless protected (by `{ }` or `" "` or `\space`) but filenames with a space are hopefully rare.

```

222 </2ekernel>
223 <*2ekernel | latexrelease>
224 <latexrelease> \IncludeInRelease{2019/10/01}%
225 <latexrelease>          {\set@curr@file}{Quote file names}%
226 \def\set@curr@file#1{%
227   \begingroup
228     \escapechar\m@ne
229     \xdef\@curr@file{%
230       \expandafter\expandafter\expandafter\unquote@name
231       \expandafter\expandafter\expandafter{%
232         \expandafter\string
233         \csname\@firstofone#1\@empty\endcsname}}%
234   \endgroup
235 }

```

```

\quote@name Quoting spaces
\quote@@name
\unquote@name
a b c      -> "a b c"
"a b c"    -> "a b c"
a" "b" "c -> "a b c"
           -> ""

```

```

236 \def\quote@name#1{"\quote@@name#1@gobble"}
237 \def\quote@@name#1{"#1\quote@@name}

```

and removing quotes ...

```

238 \def\unquote@name#1{\quote@@name#1@gobble}

```

`\IfFileExists`

```
239 \DeclareRobustCommand\IfFileExists[1]{%
240   \set@curr@file{#1}%
241   \expandafter\IfFileExists@\expandafter{\@curr@file}}
```

`\IfFileExists@` Argument #1 is `\@curr@file` so catcode 12 string with no quotes.

```
242 \long\def \IfFileExists@#1#2#3{%
243   \openin\@inputcheck"#1" %
244   \ifeof\@inputcheck
245     \ifx\input@path\@undefined
246       \def\reserved@a{#3}%
247     \else
248       \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
249     \fi
250   \else
251     \closein\@inputcheck
252     \edef\@filef@und{"#1" }%
253     \def\reserved@a{#2}%
254   \fi
255   \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`.

```
256 \long\def\@iffileonpath#1{%
257   \let\reserved@a\@secondoftwo
258   \expandafter\@tfor\expandafter\reserved@b\expandafter
259     :\expandafter=\input@path\do{%
260     \openin\@inputcheck\expandafter\quote@name\expandafter{\reserved@b#1} %
261     \ifeof\@inputcheck\else
262       \edef\@filef@und{\expandafter\quote@name\expandafter{\reserved@b#1} }%
263       \let\reserved@a\@firstoftwo%
264       \closein\@inputcheck
265       \@break@tfor
266     \fi}%
267   \reserved@a}
```

```
268 </2ekernel | latexrelease>
269 <latexrelease>\EndIncludeInRelease
270 <latexrelease>\IncludeInRelease{0000/00/00}%
271 <latexrelease>          {\set@curr@file}{Quote file names}%
272 <latexrelease>
273 <latexrelease>\let\quote@name\@undefined
274 <latexrelease>\let\quote@@name\@undefined
275 <latexrelease>\let\unquote@name\@undefined
276 <latexrelease>\let\set@curr@file\@undefined
277 <latexrelease>
278 <latexrelease>\let\IfFileExists@\@undefined
279 <latexrelease>
280 <latexrelease>\long\def \IfFileExists#1#2#3{%
281 <latexrelease>   \openin\@inputcheck#1 %
282 <latexrelease>   \ifeof\@inputcheck
283 <latexrelease>     \ifx\input@path\@undefined
284 <latexrelease>       \def\reserved@a{#3}%
285 <latexrelease>     \else
```

```

286 \latexrelease \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
287 \latexrelease \fi
288 \latexrelease \else
289 \latexrelease \closein\@inputcheck
290 \latexrelease \edef\@filef@und{#1 }%
291 \latexrelease \def\reserved@a{#2}%
292 \latexrelease \fi
293 \latexrelease \reserved@a}
294 \latexrelease
295 \latexrelease\long\def\@iffileonpath#1{%
296 \latexrelease \let\reserved@a\@secondoftwo
297 \latexrelease \expandafter\@tfor\expandafter\reserved@b\expandafter
298 \latexrelease : \expandafter=\input@path\do{%
299 \latexrelease \openin\@inputcheck\reserved@b#1 %
300 \latexrelease \ifeof\@inputcheck\else
301 \latexrelease \edef\@filef@und{\reserved@b#1 }%
302 \latexrelease \let\reserved@a\@firstoftwo%
303 \latexrelease \closein\@inputcheck
304 \latexrelease \@break@tfor
305 \latexrelease \fi}%
306 \latexrelease \reserved@a}
307 \latexrelease
308 \latexrelease\EndIncludeInRelease
309 \ifx\kernel

```

\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’.

```

310 \ifx\kernel
311 \latexrelease\IncludeInRelease{2019/10/01}%
312 \latexrelease {\InputIfFileExists}{Don't lose the file name}%
313 \ifx\kernel | \latexrelease
314 % \begin{macrocode}
315 \DeclareRobustCommand \InputIfFileExists[2]{%
316 \IfFileExists{#1}%
317 {%
318 \expandafter\@swaptwoargs\expandafter
319 {\@filef@und}{#2\@addtofilelist{#1}\@input}}

```

\@swaptwoargs Swap two arguments and return them unbraced (like `\@firstoftwo` etc).

```

320 \long\def\@swaptwoargs#1#2{#2#1}
321 \ifx\kernel | \latexrelease
322 \latexrelease\EndIncludeInRelease
323 \latexrelease\IncludeInRelease{0000/00/00}%
324 \latexrelease {\InputIfFileExists}{Don't lose the file name}%
325 \latexrelease\long\def \InputIfFileExists#1#2{%
326 \latexrelease \IfFileExists{#1}%
327 \latexrelease {\#2\@addtofilelist{#1}\@input \@filef@und}}
328 \latexrelease
329 \latexrelease\let\@swaptwoargs\@undefined
330 \latexrelease\EndIncludeInRelease
331 \ifx\kernel

```

\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^AT_EX).

```

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

\@iinput Define \@iinput (i.e., \input) in terms of \InputIfFileExists.
333 \kernel
334 \*kernel | latexrelease
335 \latexrelease\IncludeInRelease{2019/10/01}%
336 \latexrelease\@iinput{Quote file names}%
337 \def\@iinput#1{%
338   \InputIfFileExists{#1}{}%
339   {\filename@parse\@curr@file
340    \edef\reserved@a{\noexpand\@missingfileerror
341     {\filename@area\filename@base}%
342     {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
343    \reserved@a}}
344 \kernel | latexrelease
345 \latexrelease\EndIncludeInRelease
346 \latexrelease\IncludeInRelease{0000/00/00}%
347 \latexrelease\@iinput{Quote file names}%
348 \latexrelease\def\@iinput#1{%
349 \latexrelease\@InputIfFileExists{#1}{}%
350 \latexrelease\{\filename@parse{#1}%
351 \latexrelease\@edef\reserved@a{\noexpand\@missingfileerror
352 \latexrelease\{\filename@area\filename@base}%
353 \latexrelease\{\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
354 \latexrelease\@reserved@a}}
355 \latexrelease\EndIncludeInRelease
356 \*kernel

\@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, some-
thing like .bbl probably should be listed and thus should be implemented not by
\@input.
357 \def\@input#1{%
358   \IfFileExists{#1}{\@input\@filef@und}{\typeout{No file #1.}}}

\@input@ Version of \@input that does add the file to \@filelist.
359 \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.
360 \gdef\@missingfileerror#1#2{%
361   \typeout{^^J! LaTeX Error: File ‘#1.#2’ not found.^^J^^J%
362   Type X to quit or <RETURN> to proceed,^^J%
363   or enter new name. (Default extension: #2)^^J}%
364   \message{Enter file name: }%
365   {\endlinechar\m@ne
366    \global\read\m@ne to\@gtempa}%
367   \ifx\@gtempa\@empty
368   \else
369     \def\reserved@a{x}\ifx\reserved@a\@gtempa\batchmode\@end\fi
370     \def\reserved@a{X}\ifx\reserved@a\@gtempa\batchmode\@end\fi
371     \filename@parse\@gtempa

```

```

372 \edef\filename@ext{%
373 \ifx\filename@ext\relax#2\else\filename@ext\fi}%
374 \edef\reserved@a{%
375 \noexpand\inputIfFileExists
376 {\filename@area\filename@base.\filename@ext}%
377 }%
378 {\noexpand\@missingfileerror
379 {\filename@area\filename@base}{\filename@ext}}}%
380 \reserved@a
381 \fi}

```

\@obsoletefile For compatibility with L^AT_EX 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.

```

382 \def\@obsoletefile#1#2{%
383 \latex@warning@no@line{inputting ‘#1’ instead of obsolete ‘#2’}}
384 \@onlypreamble\@obsoletefile

```

19.2 Listing files

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

```

385 \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.

```

386 %\def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}

```

\listfiles A preamble command to cause `\end{document}` to list files input from the main file.

```

387 \def\listfiles{%
388 \let\listfiles\relax
389 \def\@listfiles##1##2##3##4##5##6##7##8##9\@{%
390 \def\reserved@d{\}%
391 \tfor\reserved@c:=##1##2##3##4##5##6##7##8\do{%
392 \ifx\reserved@c\reserved@d
393 \edef\filename@area{ \filename@area}%
394 \fi}}%

395 \def\@dofilelist{%
396 \typeout{^^J *File List*}%
397 \@for\@currname:=\@filelist\do{%
398 \filename@parse\@currname
399 \edef\reserved@a{%
400 \filename@base.%
401 \ifx\filename@ext\relax tex\else\filename@ext\fi}%
402 \expandafter\let\expandafter\reserved@b
403 \csname ver@\reserved@a\endcsname
404 \expandafter\expandafter\expandafter\@listfiles\expandafter
405 \filename@area\filename@base\////////////////////////////////\@
406 \typeout{%
407 \filename@area\reserved@a
408 \ifx\reserved@b\relax\else\@spaces\reserved@b\fi}}%
409 \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}

410 \@onlypreamble\listfiles

\@dofilelist

411 \let\@dofilelist\relax

412 \</2kernel>
```


File 1

ltoutenc.dtx

20 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`.

20.1 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 \LaTeX 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^AT_EX 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}
```

20.2 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.

20.3 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.
TU	generates <code>tuenc.def</code> for Unicode font encoding.
package	generates <code>fontenc.sty</code> for selecting encodings.
2kernel	for the kernel commands.

20.4 Definitions for the kernel

20.4.1 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 (*2kernel)
2 \message{font encodings,}

Far too many macros in one block here!
```

```
\DeclareTextCommand If you say:
\ProvideTextCommand
\DeclareTextSymbol
\@dec@text@cmd then \foo is defined to be \T1-cmd \foo \T1\foo, where \T1\foo is one control
\chardef@text@cmd sequence, not two! We then call \newcommand to define \T1\foo.
\@changed@cmd
\@changed@x
\TextSymbolUnavailable
\@inmathwarn

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 reimplementaion 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}%
```

The accent primitive doesn't allow things `\begingroup` to interfere between accent and base character. Therefore we need to avoid that (they are some hidden inside `\maybe@load@fontshape`). As we don't have to load the fontshape in this case (as that happened in the box above if necessary, we simply disable that part of the code temporarily. We also ignore `\ignorespaces` which has the same issue and may show up as part of `\normalfont` if that is used.

```
71 \let\maybe@load@fontshape\relax
72 \let\ignorespaces\relax
73 \accent#1 #2\egroup\spacefactor\accent@spacefactor}
```

Default definition for `\accent@spacefactor` prevents a horrible death of the above macro inside an unprotected `\edef`.

```
74 \let\accent@spacefactor\relax
```

`\hmode@bgroup`

```
75 \def\hmode@bgroup{\leavevmode\bgroup}
```

`\DeclareTextCompositeCommand` Another amusing game to play with `\expandafter`, `\csname`, 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`.

```
76 \</2kernel>
77 \<latexrelease>\IncludeInRelease{2017/04/15}{\DeclareTextCompositeCommand}
78 \<latexrelease> {test for undeclared accent}%
79 \<*2kernel|latexrelease>
80 \def\DeclareTextCompositeCommand#1#2#3#4{%
81 \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
82 \ifx\reserved@a\relax
83 \DeclareTextCommand#1{#2}{%
84 \@latex@error{\string#1 undeclared in encoding #2}\@eha}%
85 \@latex@info{Composite with undeclared \string#1 in encoding #2}%
86 \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
87 \fi
88 \expandafter\expandafter\expandafter\ifx
89 \expandafter\@car\reserved@a\relax\relax\@nil \@text@composite \else
90 \edef\reserved@b##1{%
91 \def\expandafter\noexpand
92 \csname#2\string#1\endcsname###1{%
93 \noexpand\@text@composite
```



```

94         \expandafter\noexpand\csname#2\string#1\endcsname
95         #####1\noexpand\@empty\noexpand\@text@composite
96         {##1}}}%
97     \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
98     \fi
99     \expandafter\def\csname\expandafter\string\csname
100     #2\endcsname\string#1-\string#3\@empty\endcsname{#4}%
101 }
102 \if2ekernel\latexrelease
103 \latexrelease\EndIncludeInRelease
104 \latexrelease\IncludeInRelease{0000/00/00}{\DeclareTextCompositeCommand}
105 \latexrelease{test for undeclared accent}%
106 \latexrelease\def\DeclareTextCompositeCommand#1#2#3#4{%
107 \latexrelease \expandafter\let\expandafter\reserved@a
108 \latexrelease \csname#2\string#1\endcsname
109 \expandafter\expandafter\expandafter\ifx
110 \latexrelease \expandafter\@car\reserved@a\relax\relax\@nil
111 \latexrelease \@text@composite \else
112 \latexrelease \edef\reserved@b##1{%
113 \latexrelease \def\expandafter\noexpand
114 \latexrelease \csname#2\string#1\endcsname#####1{%
115 \latexrelease \noexpand\@text@composite
116 \latexrelease \expandafter\noexpand\csname#2\string#1\endcsname
117 \latexrelease #####1\noexpand\@empty\noexpand\@text@composite
118 \latexrelease {##1}}}%
119 \latexrelease \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
120 \latexrelease \fi
121 \latexrelease \expandafter\def\csname\expandafter\string\csname
122 \latexrelease #2\endcsname\string#1-\string#3\@empty\endcsname{#4}}
123 \latexrelease\EndIncludeInRelease
124 \if2ekernel
125 \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 csname. This is so that `\'\{ \textit{e} \}` will work—it checks whether `\T1\'-\textit` 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):

```
\csname \string \T1\' - \string { \itshape e } \@empty \endcsname
```

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.

```
126 \def\@text@composite#1#2#3\@text@composite{%
127   \expandafter\@text@composite@x
128     \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.

```
129 \def\@text@composite@x#1{%
130   \ifx#1\relax
131     \expandafter\@secondoftwo
132   \else
133     \expandafter\@firstoftwo
134   \fi
135   #1}
```

The command `\DeclareTextComposite` uses `\DeclareTextCompositeCommand` to declare a command which expands out to a single glyph.

```
136 \catcode\z@=11\relax

137 \def\DeclareTextComposite#1#2#3#4{%
138   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
139   \bgroup
140     \lccode\z@#4%
141     \lowercase{%
142   \egroup
143     \reserved@a ^^@}}

144 \catcode\z@=15\relax

145 \@onlypreamble\DeclareTextComposite

146 </2ekernel>
147 <*2ekernel | latexrelease>
148 <latexrelease>\IncludeInRelease{2019/10/01}%
149 <latexrelease>          {\UseTextAccent}{Make commands robust}}%
```

`\UseTextAccent` These fragile commands access glyphs from different encodings. They use grotty
`\UseTextSymbol` low-level calls to the font selection scheme for speed, and in order to make sure
`\@use@text@encoding` that `\UseTextSymbol` doesn't do anything which you're not allowed to do between
 an `\accent` and its glyph.

For a detailed discussion of this reimplementaion and its deficiencies, see [pr/3160](#).

```
150 \DeclareRobustCommand\UseTextAccent[3]{%
151   \hmode@start@before@group
152   {%
```

Turn off the group in `\UseTextSymbol` in case this is used inside the arguments of `\UseTextAccent`.

```

153 \let\hmode@start@before@group\@firstofone
154 \let\@curr@enc\cf@encoding
155 \@use@text@encoding{#1}%
156 #2{\@use@text@encoding\@curr@enc#3}%
157 }}

158 \DeclareRobustCommand\UseTextSymbol[2]{%
159 \hmode@start@before@group
160 {%
161 \def\@wrong@font@char{\MessageBreak
162 for \noexpand\symbol'\string#2'}%
163 \@use@text@encoding{#1}%
164 #2%
165 }%
166 }

167 \ifkernel\latexrelease
168 \ifkernel\EndIncludeInRelease
169 \ifkernel\IncludeInRelease{0000/00/00}%
170 \ifkernel\{UseTextAccent}{Make commands robust}%
171 \ifkernel
172 \ifkernel\kernel@make@fragile\UseTextAccent
173 \ifkernel\kernel@make@fragile\UseTextSymbol
174 \ifkernel
175 \ifkernel\EndIncludeInRelease
176 \ifkernel

```

Switch to a different text encoding without any grouping for use in `\UseTextAccent` or `\UseTextSymbol` (and for `\oldstylenums`).

```

177 \def\@use@text@encoding#1{%
178 \edef\@f@encoding{#1}%
179 \xdef\@font@name{%
180 \csname\curr@fontshape/\@f@size\endcsname}%
181 \pickup@font
182 \font@name
183 \@@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.

```

184 \let\hmode@start@before@group\leavevmode

```

`\DeclareTextSymbolDefault` Some syntactic sugar. Again, these should probably be optimized for speed.

```

\DeclareTextAccentDefault
185 \def\DeclareTextSymbolDefault#1#2{%
186 \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}

187 \def\DeclareTextAccentDefault#1#2{%
188 \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}

189 \onlypreamble\DeclareTextSymbolDefault
190 \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.

```
191 \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).

```
192   \expandafter\ifx\csname#2\string#1\endcsname\relax
193   \else
```

Else: throw away that declaration.

```
194     \global\expandafter\let\csname#2\string#1\endcsname
195     \@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:

```
196   \expandafter\expandafter\expandafter
197   \ifx\expandafter\@thirdofthree#1\@undefined
198     \expandafter\gdef\expandafter#1\expandafter
199       {\csname ?-cmd\expandafter\endcsname\expandafter
200        #1\csname?\string#1\endcsname}%
201   \fi
202 \fi
203 }
204 \@onlypreamble\UndeclareTextCommand
```

20.4.2 Hyphenation

```
\patterns We redefine \patterns and \hyphenation to allow the use of commands declared
\@patterns with \DeclareText* to be used inside them.
\hyphenation
\@hyphenation 205 %\let\@patterns\patterns
                206 %\let\@hyphenation\hyphenation
                207 %\def\patterns{%
208 %   \bgroup
209 %     \let\protect\@empty
210 %     \let\@typeset@protect\@empty
211 %     \let\@changed@x\@changed@x@mouth
212 %   \afterassignment\egroup
213 %   \@patterns
214 %}
215 %\def\hyphenation{%
```

```

216 % \bgroup
217 % \let\protect\@empty
218 % \let\@typeset@protect\@empty
219 % \let\@changed@x\@changed@x@mouth
220 % \afterassignment\egroup
221 % \@@hyphenation
222 %}

```

20.4.3 Miscellania

`\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.

```

223 \def\@tabacckludge#1{\expandafter\@changed@cmd
224 \csname\string#1\endcsname\relax}
225 \let\a=\@tabacckludge

```

20.4.4 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_EX 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:

```

226 \DeclareTextAccentDefault{"}{OT1}
227 \DeclareTextAccentDefault{'}{OT1}
228 \DeclareTextAccentDefault{.}{OT1}
229 \DeclareTextAccentDefault{=}{OT1}
230 \DeclareTextAccentDefault{H}{OT1}
231 \DeclareTextAccentDefault{^}{OT1}
232 \DeclareTextAccentDefault{'}{OT1}
233 \DeclareTextAccentDefault{b}{OT1}
234 \DeclareTextAccentDefault{c}{OT1}
235 \DeclareTextAccentDefault{d}{OT1}
236 \DeclareTextAccentDefault{r}{OT1}
237 \DeclareTextAccentDefault{u}{OT1}
238 \DeclareTextAccentDefault{v}{OT1}

```

```
239 \DeclareTextAccentDefault{\~}{OT1}
```

Some symbols from OT1:

```
240 %\DeclareTextSymbolDefault{\AA}{OT1}
```

```
241 \DeclareTextSymbolDefault{\AE}{OT1}
```

```
242 \DeclareTextSymbolDefault{\L}{OT1}
```

```
243 \DeclareTextSymbolDefault{\OE}{OT1}
```

```
244 \DeclareTextSymbolDefault{\O}{OT1}
```

```
245 %\DeclareTextSymbolDefault{\aa}{OT1}
```

```
246 \DeclareTextSymbolDefault{\ae}{OT1}
```

```
247 \DeclareTextSymbolDefault{\i}{OT1}
```

```
248 \DeclareTextSymbolDefault{\j}{OT1}
```

```
249 \DeclareTextSymbolDefault{\ij}{OT1}
```

```
250 \DeclareTextSymbolDefault{\IJ}{OT1}
```

```
251 \DeclareTextSymbolDefault{\l}{OT1}
```

```
252 \DeclareTextSymbolDefault{\oe}{OT1}
```

```
253 \DeclareTextSymbolDefault{\o}{OT1}
```

```
254 \DeclareTextSymbolDefault{\ss}{OT1}
```

```
255 \DeclareTextSymbolDefault{\textdollar}{OT1}
```

```
256 \DeclareTextSymbolDefault{\textemdash}{OT1}
```

```
257 \DeclareTextSymbolDefault{\textendash}{OT1}
```

```
258 \DeclareTextSymbolDefault{\textexclamdown}{OT1}
```

```
259 %\DeclareTextSymbolDefault{\texthyphenchar}{OT1}
```

```
260 %\DeclareTextSymbolDefault{\texthyphen}{OT1}
```

```
261 \DeclareTextSymbolDefault{\textquestiondown}{OT1}
```

```
262 \DeclareTextSymbolDefault{\textquotedblleft}{OT1}
```

```
263 \DeclareTextSymbolDefault{\textquotedblright}{OT1}
```

```
264 \DeclareTextSymbolDefault{\textquoteleft}{OT1}
```

```
265 \DeclareTextSymbolDefault{\textquoteright}{OT1}
```

```
266 \DeclareTextSymbolDefault{\textsterling}{OT1}
```

Some symbols from OMS:

```
267 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
```

```
268 \DeclareTextSymbolDefault{\textbackslash}{OMS}
```

```
269 \DeclareTextSymbolDefault{\textbar}{OMS}
```

```
270 \DeclareTextSymbolDefault{\textbardbl}{OMS}
```

```
271 \DeclareTextSymbolDefault{\textbraceleft}{OMS}
```

```
272 \DeclareTextSymbolDefault{\textbraceright}{OMS}
```

```
273 \DeclareTextSymbolDefault{\textbullet}{OMS}
```

```
274 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
```

```
275 \DeclareTextSymbolDefault{\textdagger}{OMS}
```

```
276 \DeclareTextSymbolDefault{\textparagraph}{OMS}
```

```
277 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
```

```
278 \DeclareTextSymbolDefault{\textsection}{OMS}
```

```
279 \DeclareTextAccentDefault{\textcircled}{OMS}
```

Some symbols from OML:

```
280 \DeclareTextSymbolDefault{\textless}{OML}
```

```
281 \DeclareTextSymbolDefault{\textgreater}{OML}
```

```
282 \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.

```

283 \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
284 % \expandafter\def\expandafter
285 % \copyright\expandafter{\expandafter{\copyright}}

286 \DeclareTextCommandDefault{\textasciicircum}{\textasciicircum}
287 \DeclareTextCommandDefault{\textasciitilde}{\textasciitilde}
288 \DeclareTextCommandDefault{\textunderscore}{\textunderscore}
289 \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

```

There is no good reason anymore to fake `\textcompwordmark`.

```

290 %\DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
291 \DeclareTextSymbolDefault{\textcompwordmark}{T1}

292 \DeclareTextCommandDefault{\textvisiblespace}{%
293   \mbox{\kern.06em\vrule \@height.3ex}%
294   \vbox{\hrule \@width.3em}%
295   \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).

```

296 \DeclareTextCommandDefault{\textellipsis}{%
297   .\kern\fontdimen3\font
298   .\kern\fontdimen3\font
299   .\kern\fontdimen3\font}

300 %\DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape r}}
301 \DeclareTextCommandDefault{\textregistered}{\textcircled{%
302   \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
303 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
304 \DeclareTextCommandDefault{\SS}{SS}

305 \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
306 \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}

```

20.4.5 Math material

Some commands can be used in both text and math mode:

```

307 \DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
308 \DeclareRobustCommand{\{ }{\ifmmode\lbrace\else\textbraceleft\fi}
309 \DeclareRobustCommand{\} }{\ifmmode\rbrace\else\textbraceright\fi}
310 \DeclareRobustCommand{\P }{\ifmmode\mathparagraph\else\textparagraph\fi}
311 \DeclareRobustCommand{\S }{\ifmmode\mathsection\else\textsection\fi}
312 \DeclareRobustCommand{\dag }{\ifmmode\dagger\else\textdagger\fi}
313 \DeclareRobustCommand{\ddag }{\ifmmode\ddagger\else\textdaggerdbl\fi}

```

For historical reasons `\copyright` needs `{ }` around the definition in maths.

```

314 \DeclareRobustCommand{\_ }{%
315   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
316 \DeclareRobustCommand{\copyright }{%
317   \ifmmode\nfss@text{\textcopyright}}\else\textcopyright\fi}
318 \DeclareRobustCommand{\pounds }{%
319   \ifmmode\mathsterling\else\textsterling\fi}

320 \DeclareRobustCommand{\dots }{%
321   \ifmmode\mathellipsis\else\textellipsis\fi}

```

322 \let\ldots\dots

Default definition of the commabelow accent.

```

323 \endkernel
324 \latexrelease\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%
325 \*2ekernel | latexrelease
326 \DeclareTextCommandDefault\textcommabelow[1]
327   {\hmode\bgroup\ooalign{\null#1\crr\hidewidth\raise-.31ex
328     \hbox{\check@mathfonts\fontsize\ssf@size\z@
329       \math@fontsfalse\selectfont,}\hidewidth}\egroup}
330 \latexrelease\EndIncludeInRelease
331 \endkernel | latexrelease
332 \latexrelease\IncludeInRelease{0000/00/00}{\textcommabelow}{comma accent}%
333 \latexrelease\let\textcommabelow\undefined
334 \latexrelease\expandafter
335 \latexrelease\let\csname\string\T1\string\c-G\endcsname\undefined
336 \latexrelease\expandafter
337 \latexrelease\let\csname\string\T1\string\c-K\endcsname\undefined
338 \latexrelease\expandafter
339 \latexrelease\let\csname\string\T1\string\c-k\endcsname\undefined
340 \latexrelease\expandafter
341 \latexrelease\let\csname\string\T1\string\c-L\endcsname\undefined
342 \latexrelease\expandafter
343 \latexrelease\let\csname\string\T1\string\c-l\endcsname\undefined
344 \latexrelease\expandafter
345 \latexrelease\let\csname\string\T1\string\c-N\endcsname\undefined
346 \latexrelease\expandafter
347 \latexrelease\let\csname\string\T1\string\c-n\endcsname\undefined
348 \latexrelease\expandafter
349 \latexrelease\let\csname\string\T1\string\c-R\endcsname\undefined
350 \latexrelease\expandafter
351 \latexrelease\let\csname\string\T1\string\c-r\endcsname\undefined
352 \latexrelease\EndIncludeInRelease

```

Default definition of the commaabove accent(E.G.).

```

353 \latexrelease\IncludeInRelease{2016/02/01}{\textcommaabove}{comma above}%
354 \*2ekernel | latexrelease
355 \DeclareTextCommandDefault\textcommaabove[1]{%
356   \hmode\bgroup
357   \ooalign{%
358     \hidewidth
359     \raise.7ex\hbox{%
360       \check@mathfonts\fontsize\ssf@size\z@\math@fontsfalse\selectfont'%
361     }%
362     \hidewidth\crr
363     \null#1\crr
364   }%
365   \egroup
366 }
367 \latexrelease\EndIncludeInRelease
368 \endkernel | latexrelease
369 \latexrelease\IncludeInRelease{0000/00/00}{\textcommaabove}{comma above}%
370 \latexrelease\let\textcommaabove\undefined
371 \latexrelease\expandafter
372 \latexrelease\let\csname\string\OT1\string\c-g\endcsname\undefined

```



```

373 \latexrelease\expandafter
374 \latexrelease\let\csname\string\T1\string\c-g\endcsname\@undefined
375 \latexrelease\EndIncludeInRelease

```

20.5 Definitions for the OT1 encoding

The definitions for the ‘T_EX text’ (OT1) encoding.

Declare the encoding.

```

376 (*OT1)
377 \DeclareFontEncoding{OT1}{-}{-}

```

Declare the accents.

```

378 \DeclareTextAccent{"}{OT1}{127}
379 \DeclareTextAccent{'}{OT1}{19}
380 \DeclareTextAccent{.}{OT1}{95}
381 \DeclareTextAccent{=}{OT1}{22}
382 \DeclareTextAccent{~}{OT1}{94}
383 \DeclareTextAccent{\'}{OT1}{18}
384 \DeclareTextAccent{\~}{OT1}{126}
385 \DeclareTextAccent{\H}{OT1}{125}
386 \DeclareTextAccent{\u}{OT1}{21}
387 \DeclareTextAccent{\v}{OT1}{20}
388 \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.

```

389 \DeclareTextCommand{\b}{OT1}[1]
390   {\hmode\bgroup\o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-3ex}%
391     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
392 \DeclareTextCommand{\c}{OT1}[1]
393   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
394     \else{\oalign{\unhbox\z@\cr\hidewidth\char24\hidewidth}}\fi}
395 \DeclareTextCommand{\d}{OT1}[1]
396   {\hmode\bgroup
397     \o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

398 \DeclareTextSymbol{\AE}{OT1}{29}
399 \DeclareTextSymbol{\OE}{OT1}{30}
400 \DeclareTextSymbol{\O}{OT1}{31}
401 \DeclareTextSymbol{\ae}{OT1}{26}
402 \DeclareTextSymbol{\i}{OT1}{16}
403 \DeclareTextSymbol{\j}{OT1}{17}
404 \DeclareTextSymbol{\oe}{OT1}{27}
405 \DeclareTextSymbol{\o}{OT1}{28}
406 \DeclareTextSymbol{\ss}{OT1}{25}
407 \DeclareTextSymbol{\textendash}{OT1}{124}
408 \DeclareTextSymbol{\textendash}{OT1}{123}

```

Using the ligatures helps with OT1 fonts that have `\textexclamdown` and `\textquestiondown` in unusual positions.

```

409 %\DeclareTextSymbol{\textexclamdown}{OT1}{60}
410 %\DeclareTextSymbol{\textquestiondown}{OT1}{62}
411 \DeclareTextCommand{\textexclamdown}{OT1}{!}

```

```

412 \DeclareTextCommand{\textquestiondown}{OT1}{?'}
413 %\DeclareTextSymbol{\textthyphenchar}{OT1}{'\-}
414 %\DeclareTextSymbol{\textthyphen}{OT1}{'\-}
415 \DeclareTextSymbol{\textquotedblleft}{OT1}{92}
416 \DeclareTextSymbol{\textquotedblright}{OT1}{'\'}
417 \DeclareTextSymbol{\textquoteleft}{OT1}{'\'}
418 \DeclareTextSymbol{\textquoteright}{OT1}{'\'}

```

Some symbols which are faked from others:

```

419 % \DeclareTextCommand{\aa}{OT1}
420 %     {\accent23a}
421 \DeclareTextCommand{\L}{OT1}
422     {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
423 \DeclareTextCommand{\l}{OT1}
424     {\hmode\bgroup\@xxxii l\egroup}
425 % \DeclareTextCommand{\AA}{OT1}
426 %     {\leavevmode\setbox\z@\hbox{h}\dimen@ht\z@\advance\dimen@-1ex%
427 %     \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`.

```

428 \DeclareTextCompositeCommand{\r}{OT1}{A}
429     {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
430     \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. Therefore we fake it for the OT1 encoding.

```

431 \DeclareTextCommand{\ij}{OT1}{%
432     \nobreak\hskip\z@skip i\kern-0.02em j\nobreak\hskip\z@skip}
433 \DeclareTextCommand{\IJ}{OT1}{%
434     \nobreak\hskip\z@skip I\kern-0.02em J\nobreak\hskip\z@skip}

```

In the OT1 encoding, £ and \$ share a slot.

```

435 \DeclareTextCommand{\textdollar}{OT1}{\hmode\bgroup
436     \ifdim \fontdimen\@ne\font >\z@
437         \slshape
438     \else
439         \upshape
440     \fi
441     \char'\$\egroup}
442 \DeclareTextCommand{\textsterling}{OT1}{\hmode\bgroup
443     \ifdim \fontdimen\@ne\font >\z@
444         \itshape
445     \else
446         \fontshape{ui}\selectfont
447     \fi
448     \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^AT_EX 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.

```

449 \DeclareTextComposite{\.}{OT1}{i}{'\i}
450 \DeclareTextComposite{\.}{OT1}{i}{'\i}

```

```

451 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
452 \DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
453 \DeclareTextCompositeCommand{\^}{OT1}{i}{\^'\i}
454 \DeclareTextCompositeCommand{\"}{OT1}{i}{\"'\i}

```

T1 encoding is given more extensive set of overloads for \c But here we just adjust \c{g}.

```

455 \ifx\textcommaabove\undefined\else
456 \DeclareTextCompositeCommand{\c}{OT1}{g}{\textcommaabove{g}}
457 \fi
458 </OT1>

```

20.6 Definitions for the T1 encoding

The definitions for the ‘Extended T_EX text’ (T1) encoding.

Declare the encoding.

```

459 (*T1)
460 \DeclareFontEncoding{T1}{-}{-}

```

Declare the accents.

```

461 \DeclareTextAccent{\'}{T1}{0}
462 \DeclareTextAccent{\'}{T1}{1}
463 \DeclareTextAccent{\^}{T1}{2}
464 \DeclareTextAccent{\^}{T1}{3}
465 \DeclareTextAccent{\"}{T1}{4}
466 \DeclareTextAccent{\H}{T1}{5}
467 \DeclareTextAccent{\r}{T1}{6}
468 \DeclareTextAccent{\v}{T1}{7}
469 \DeclareTextAccent{\u}{T1}{8}
470 \DeclareTextAccent{\=}{T1}{9}
471 \DeclareTextAccent{\.}{T1}{10}

```

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.

```

472 \DeclareTextCommand{\b}{T1}[1]
473   {\hmode\bgroup\o@lign{\relax#1\crrcr\hidewidth\ltx@sh@ft{-3ex}%
474     \vbox to.2ex{\hbox{\char9}\vss}\hidewidth}\egroup}
475 \DeclareTextCommand{\c}{T1}[1]
476   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent11 #1%
477     \else{\oalign{\unhbox\z@\crrcr
478       \hidewidth\char11\hidewidth}}\fi}
479 \DeclareTextCommand{\d}{T1}[1]
480   {\hmode\bgroup
481     \o@lign{\relax#1\crrcr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
482 \DeclareTextCommand{\k}{T1}[1]
483   {\hmode\bgroup\oalign{\null#1\crrcr\hidewidth\char12}\egroup}
484 \DeclareTextCommand{\textogonekcentered}{T1}[1]
485   {\hmode\bgroup\oalign{%
486     \null#1\crrcr\hidewidth\char12\hidewidth}\egroup}

```

Some symbols are constructed.

Slot 24 contains a small circle intended for construction of these two glyphs.

```

487 \DeclareTextCommand{\textperthousand}{T1}

```

```

488 {\%\char 24 } % space or ‘relax as delimiter?
489 \DeclareTextCommand{\textpertenthousand}{T1}
490 {\%\char 24\char 24 } % space or ‘relax as delimiter?

```

For Maltese, \Hwithstroke and \hwithstroke are needed.

```

491 \DeclareTextCommand{\Hwithstroke}{T1}
492 {%
493   \hmode\bgroup
494   \vphantom{H}%
495   \sbox\z@{H}%
496   \ooalign{%
497     H\cr
498     \hidewidth
499     \vrule
500     height \dimexpr 0.7\ht\z@+0.1ex\relax
501     depth  -0.7\ht\z@
502     width  0.8\wd\z@
503     \hidewidth\cr
504   }%
505   \egroup
506 }
507 \DeclareTextCommand{\hwithstroke}{T1}
508 {%
509   \hmode\bgroup
510   \vphantom{h}%
511   \sbox\z@{h}%
512   \ooalign{%
513     h\cr
514     \kern0.075\wd\z@
515     \vrule
516     height \dimexpr 0.7\ht\z@+0.1ex\relax
517     depth  -0.7\ht\z@
518     width  0.4\wd\z@
519     \hidewidth\cr
520   }%
521   \egroup
522 }

```

Declare the text symbols.

```

523 %\DeclareTextSymbol{\AA}{T1}{197}
524 \DeclareTextSymbol{\AE}{T1}{198}
525 \DeclareTextSymbol{\DH}{T1}{208}
526 \DeclareTextSymbol{\DJ}{T1}{208}
527 \DeclareTextSymbol{\L}{T1}{138}
528 \DeclareTextSymbol{\NG}{T1}{141}
529 \DeclareTextSymbol{\OE}{T1}{215}
530 \DeclareTextSymbol{\O}{T1}{216}
531 \DeclareTextSymbol{\SS}{T1}{223}
532 \DeclareTextSymbol{\TH}{T1}{222}
533 %\DeclareTextSymbol{\aa}{T1}{229}
534 \DeclareTextSymbol{\ae}{T1}{230}
535 \DeclareTextSymbol{\dh}{T1}{240}
536 \DeclareTextSymbol{\dj}{T1}{158}

```

```

537 \DeclareTextSymbol{\guillemetleft}{T1}{19}
538 \DeclareTextSymbol{\guillemetright}{T1}{20}
539 % old Adobe names
540 \DeclareTextSymbol{\guillemotleft}{T1}{19}
541 \DeclareTextSymbol{\guillemotright}{T1}{20}

542 \DeclareTextSymbol{\guilsinglleft}{T1}{14}
543 \DeclareTextSymbol{\guilsinglright}{T1}{15}
544 \DeclareTextSymbol{\i}{T1}{25}
545 \DeclareTextSymbol{\j}{T1}{26}
546 \DeclareTextSymbol{\ij}{T1}{188}
547 \DeclareTextSymbol{\IJ}{T1}{156}
548 \DeclareTextSymbol{\l}{T1}{170}
549 \DeclareTextSymbol{\ng}{T1}{173}
550 \DeclareTextSymbol{\oe}{T1}{247}
551 \DeclareTextSymbol{\o}{T1}{248}
552 \DeclareTextSymbol{\quotedblbase}{T1}{18}
553 \DeclareTextSymbol{\quotesinglbase}{T1}{13}
554 \DeclareTextSymbol{\ss}{T1}{255}
555 \DeclareTextSymbol{\textasciicircum}{T1}{'\^}
556 \DeclareTextSymbol{\textasciitilde}{T1}{'\~}
557 \DeclareTextSymbol{\textbackslash}{T1}{'\\}
558 \DeclareTextSymbol{\textbar}{T1}{'|}
559 \DeclareTextSymbol{\textbraceleft}{T1}{'\{ }
560 \DeclareTextSymbol{\textbraceright}{T1}{'\} }
561 \DeclareTextSymbol{\textcompwordmark}{T1}{23}
562 \DeclareTextSymbol{\textdollar}{T1}{'\$}
563 \DeclareTextSymbol{\textemdash}{T1}{22}
564 \DeclareTextSymbol{\textendash}{T1}{21}
565 \DeclareTextSymbol{\textexclamdown}{T1}{189}
566 \DeclareTextSymbol{\textgreater}{T1}{'\>}
567 %\DeclareTextSymbol{\textthyphenchar}{T1}{127}
568 %\DeclareTextSymbol{\textthyphen}{T1}{'\-}
569 \DeclareTextSymbol{\textless}{T1}{'\<}
570 \DeclareTextSymbol{\textquestiondown}{T1}{190}
571 \DeclareTextSymbol{\textquotedblleft}{T1}{16}
572 \DeclareTextSymbol{\textquotedblright}{T1}{17}
573 \DeclareTextSymbol{\textquotedbl}{T1}{'" }
574 \DeclareTextSymbol{\textquoteleft}{T1}{'\` }
575 \DeclareTextSymbol{\textquoteright}{T1}{'\` }
576 \DeclareTextSymbol{\textsection}{T1}{159}
577 \DeclareTextSymbol{\textsterling}{T1}{191}
578 \DeclareTextSymbol{\textunderscore}{T1}{95}
579 \DeclareTextSymbol{\textvisiblespace}{T1}{32}
580 \DeclareTextSymbol{\th}{T1}{254}

```

Declare the composites.

```

581 \DeclareTextComposite{\.}{T1}{i}{'\i}
582 \DeclareTextComposite{\.}{T1}{i}{'\i}

"80 = 128

583 \DeclareTextComposite{\u}{T1}{A}{128}
584 \DeclareTextComposite{\k}{T1}{A}{129}
585 \DeclareTextComposite{\'}{T1}{C}{130}
586 \DeclareTextComposite{\v}{T1}{C}{131}

```

```

587 \DeclareTextComposite{\v}{T1}{D}{132}
588 \DeclareTextComposite{\v}{T1}{E}{133}
589 \DeclareTextComposite{\k}{T1}{E}{134}
590 \DeclareTextComposite{\u}{T1}{G}{135}
"88 = 136
591 \DeclareTextComposite{\'}{T1}{L}{136}
592 \DeclareTextComposite{\v}{T1}{L}{137}
593 \DeclareTextComposite{\'}{T1}{N}{139}
594 \DeclareTextComposite{\v}{T1}{N}{140}
595 \DeclareTextComposite{\H}{T1}{O}{142}
596 \DeclareTextComposite{\'}{T1}{R}{143}
"90 = 144
597 \DeclareTextComposite{\v}{T1}{R}{144}
598 \DeclareTextComposite{\'}{T1}{S}{145}
599 \DeclareTextComposite{\v}{T1}{S}{146}
600 \DeclareTextComposite{\c}{T1}{S}{147}
601 \DeclareTextComposite{\v}{T1}{T}{148}
602 \DeclareTextComposite{\c}{T1}{T}{149}
603 \DeclareTextComposite{\H}{T1}{U}{150}
604 \DeclareTextComposite{\r}{T1}{U}{151}
"98 = 152
605 \DeclareTextComposite{\"}{T1}{Y}{152}
606 \DeclareTextComposite{\'}{T1}{Z}{153}
607 \DeclareTextComposite{\v}{T1}{Z}{154}
608 \DeclareTextComposite{\.}{T1}{Z}{155}
609 \DeclareTextComposite{\.}{T1}{I}{157}
"A0 = 160
610 \DeclareTextComposite{\u}{T1}{a}{160}
611 \DeclareTextComposite{\k}{T1}{a}{161}
612 \DeclareTextComposite{\'}{T1}{c}{162}
613 \DeclareTextComposite{\v}{T1}{c}{163}
614 \DeclareTextComposite{\v}{T1}{d}{164}
615 \DeclareTextComposite{\v}{T1}{e}{165}
616 \DeclareTextComposite{\k}{T1}{e}{166}
617 \DeclareTextComposite{\u}{T1}{g}{167}
"A8 = 168
618 \DeclareTextComposite{\'}{T1}{l}{168}
619 \DeclareTextComposite{\v}{T1}{l}{169}
620 \DeclareTextComposite{\'}{T1}{n}{171}
621 \DeclareTextComposite{\v}{T1}{n}{172}
622 \DeclareTextComposite{\H}{T1}{o}{174}
623 \DeclareTextComposite{\'}{T1}{r}{175}
"B0 = 176
624 \DeclareTextComposite{\v}{T1}{r}{176}
625 \DeclareTextComposite{\'}{T1}{s}{177}
626 \DeclareTextComposite{\v}{T1}{s}{178}
627 \DeclareTextComposite{\c}{T1}{s}{179}
628 \DeclareTextComposite{\v}{T1}{t}{180}
629 \DeclareTextComposite{\c}{T1}{t}{181}
630 \DeclareTextComposite{\H}{T1}{u}{182}
631 \DeclareTextComposite{\r}{T1}{u}{183}

```

```

"B8 = 184
632 \DeclareTextComposite{"}{T1}{y}{184}
633 \DeclareTextComposite{'}{T1}{z}{185}
634 \DeclareTextComposite{v}{T1}{z}{186}
635 \DeclareTextComposite{.}{T1}{z}{187}
"C0 = 192
636 \DeclareTextComposite{'}{T1}{A}{192}
637 \DeclareTextComposite{'}{T1}{A}{193}
638 \DeclareTextComposite{^}{T1}{A}{194}
639 \DeclareTextComposite{~}{T1}{A}{195}
640 \DeclareTextComposite{"}{T1}{A}{196}
641 \DeclareTextComposite{r}{T1}{A}{197}
642 \DeclareTextComposite{c}{T1}{C}{199}
"C8 = 200
643 \DeclareTextComposite{'}{T1}{E}{200}
644 \DeclareTextComposite{'}{T1}{E}{201}
645 \DeclareTextComposite{^}{T1}{E}{202}
646 \DeclareTextComposite{"}{T1}{E}{203}
647 \DeclareTextComposite{'}{T1}{I}{204}
648 \DeclareTextComposite{'}{T1}{I}{205}
649 \DeclareTextComposite{^}{T1}{I}{206}
650 \DeclareTextComposite{"}{T1}{I}{207}
"D0 = 208
651 \DeclareTextComposite{~}{T1}{N}{209}
652 \DeclareTextComposite{'}{T1}{O}{210}
653 \DeclareTextComposite{'}{T1}{O}{211}
654 \DeclareTextComposite{^}{T1}{O}{212}
655 \DeclareTextComposite{~}{T1}{O}{213}
656 \DeclareTextComposite{"}{T1}{O}{214}
"D8 = 216
657 \DeclareTextComposite{'}{T1}{U}{217}
658 \DeclareTextComposite{'}{T1}{U}{218}
659 \DeclareTextComposite{^}{T1}{U}{219}
660 \DeclareTextComposite{"}{T1}{U}{220}
661 \DeclareTextComposite{'}{T1}{Y}{221}
"E0 = 224
662 \DeclareTextComposite{'}{T1}{a}{224}
663 \DeclareTextComposite{'}{T1}{a}{225}
664 \DeclareTextComposite{^}{T1}{a}{226}
665 \DeclareTextComposite{~}{T1}{a}{227}
666 \DeclareTextComposite{"}{T1}{a}{228}
667 \DeclareTextComposite{r}{T1}{a}{229}
668 \DeclareTextComposite{c}{T1}{c}{231}
"E8 = 232
669 \DeclareTextComposite{'}{T1}{e}{232}
670 \DeclareTextComposite{'}{T1}{e}{233}
671 \DeclareTextComposite{^}{T1}{e}{234}
672 \DeclareTextComposite{"}{T1}{e}{235}
673 \DeclareTextComposite{'}{T1}{i}{236}
674 \DeclareTextComposite{'}{T1}{i}{236}

```

```

675 \DeclareTextComposite{\'}{T1}{i}{237}
676 \DeclareTextComposite{\'}{T1}{i}{237}
677 \DeclareTextComposite{\^}{T1}{i}{238}
678 \DeclareTextComposite{\^}{T1}{i}{238}
679 \DeclareTextComposite{\"}{T1}{i}{239}
680 \DeclareTextComposite{\"}{T1}{i}{239}
"FO = 240
681 \DeclareTextComposite{\~}{T1}{n}{241}
682 \DeclareTextComposite{\'}{T1}{o}{242}
683 \DeclareTextComposite{\'}{T1}{o}{243}
684 \DeclareTextComposite{\~}{T1}{o}{244}
685 \DeclareTextComposite{\~}{T1}{o}{245}
686 \DeclareTextComposite{\"}{T1}{o}{246}
"FO = 248
687 \DeclareTextComposite{\'}{T1}{u}{249}
688 \DeclareTextComposite{\'}{T1}{u}{250}
689 \DeclareTextComposite{\~}{T1}{u}{251}
690 \DeclareTextComposite{\"}{T1}{u}{252}
691 \DeclareTextComposite{\'}{T1}{y}{253}

692 \DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
693 \DeclareTextCompositeCommand{\k}{T1}{O}{\textogonekcentered{O}}

694 \ifx\textcommaabove\undefined\else
695 \DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
696 \fi
697 \ifx\textcommabelow\undefined\else
698 \DeclareTextCompositeCommand{\c}{T1}{G}{\textcommabelow{G}}
699 \DeclareTextCompositeCommand{\c}{T1}{K}{\textcommabelow{K}}
700 \DeclareTextCompositeCommand{\c}{T1}{k}{\textcommabelow{k}}
701 \DeclareTextCompositeCommand{\c}{T1}{L}{\textcommabelow{L}}
702 \DeclareTextCompositeCommand{\c}{T1}{l}{\textcommabelow{l}}
703 \DeclareTextCompositeCommand{\c}{T1}{N}{\textcommabelow{N}}
704 \DeclareTextCompositeCommand{\c}{T1}{n}{\textcommabelow{n}}
705 \DeclareTextCompositeCommand{\c}{T1}{R}{\textcommabelow{R}}
706 \DeclareTextCompositeCommand{\c}{T1}{r}{\textcommabelow{r}}
707 \fi
708 \</T1>

```

20.7 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^AT_EX text symbols.

Declare the encoding.

```

709 (*OMS)
710 \DeclareFontEncoding{OMS}{-}{-}

```

Declare the symbols. Note that slot 13 has in places been named \Orb: please root out and destroy this impolity wherever you find it!

```

711 \DeclareTextSymbol{\textasteriskcentered}{OMS}{3}    % "03
712 \DeclareTextSymbol{\textbackslash}{OMS}{110}        % "6E
713 \DeclareTextSymbol{\textbar}{OMS}{106}              % "6A

```



```

714 \DeclareTextSymbol{\textbardbl}{OMS}{107} % "6B
715 \DeclareTextSymbol{\textbraceleft}{OMS}{102} % "66
716 \DeclareTextSymbol{\textbraceright}{OMS}{103} % "67
717 \DeclareTextSymbol{\textbullet}{OMS}{15} % "0F
718 \DeclareTextSymbol{\textdaggerdbl}{OMS}{122} % "7A
719 \DeclareTextSymbol{\textdagger}{OMS}{121} % "79
720 \DeclareTextSymbol{\textparagraph}{OMS}{123} % "7B
721 \DeclareTextSymbol{\textperiodcentered}{OMS}{1} % "01
722 \DeclareTextSymbol{\textsection}{OMS}{120} % "78
723 \DeclareTextSymbol{\textbigcircle}{OMS}{13} % "0D
724 \DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
725 \ooalign{%
726 \hfil \raise .07ex\hbox {\upshape#1}\hfil \cr
727 \char 13 % "0D
728 }%
729 \egroup}
730 \end{OMS}

```

20.8 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.

```

731 \begin{OML}
732 \DeclareFontEncoding{OML}{}{}
733 \DeclareTextSymbol{\textless}{OML}{'\<}
734 \DeclareTextSymbol{\textgreater}{OML}{'\>}
735 \DeclareTextAccent{\t}{OML}{127} % "7F
736 \end{OML}

```

20.9 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.

```

737 \begin{OT4}
738 \DeclareFontEncoding{OT4}{}{}
739 \DeclareFontSubstitution{OT4}{cmr}{m}{n}
740 \DeclareTextAccent{"}{OT4}{127}
741 \DeclareTextAccent{'}{OT4}{19}
742 \DeclareTextAccent{.}{OT4}{95}
743 \DeclareTextAccent{=}{OT4}{22}
744 \DeclareTextAccent{^}{OT4}{94}

```

```

745 \DeclareTextAccent{\`}{OT4}{18}
746 \DeclareTextAccent{\~}{OT4}{126}
747 \DeclareTextAccent{\H}{OT4}{125}
748 \DeclareTextAccent{\u}{OT4}{21}
749 \DeclareTextAccent{\v}{OT4}{20}
750 \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:

```

751 \DeclareTextCommand{\k}{OT4}[1]{%
752   \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.

```

753 \DeclareTextCommand{\b}{OT4}[1]
754   {\hmode\bgroup\o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-3ex}%
755     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
756 \DeclareTextCommand{\c}{OT4}[1]
757   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
758     \else{\ooalign{\unhbox\z@\cr\hidewidth\char24\hidewidth}}\fi}
759 \DeclareTextCommand{\d}{OT4}[1]
760   {\hmode\bgroup
761     \o@lign{\relax#1\cr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

762 \DeclareTextSymbol{\AE}{OT4}{29}
763 \DeclareTextSymbol{\OE}{OT4}{30}
764 \DeclareTextSymbol{\O}{OT4}{31}
765 \DeclareTextSymbol{\L}{OT4}{138}
766 \DeclareTextSymbol{\ae}{OT4}{26}

767 \DeclareTextSymbol{\guillemetleft}{OT4}{174}
768 \DeclareTextSymbol{\guillemetright}{OT4}{175}
769 % old Adobe names
770 \DeclareTextSymbol{\guillemotleft}{OT4}{174}
771 \DeclareTextSymbol{\guillemotright}{OT4}{175}

772 \DeclareTextSymbol{\i}{OT4}{16}
773 \DeclareTextSymbol{\j}{OT4}{17}
774 \DeclareTextSymbol{\l}{OT4}{170}
775 \DeclareTextSymbol{\o}{OT4}{28}
776 \DeclareTextSymbol{\oe}{OT4}{27}
777 \DeclareTextSymbol{\quotedblbase}{OT4}{255}
778 \DeclareTextSymbol{\ss}{OT4}{25}
779 \DeclareTextSymbol{\textendash}{OT4}{124}
780 \DeclareTextSymbol{\textendash}{OT4}{123}
781 \DeclareTextSymbol{\textexclamdown}{OT4}{60}
782 %\DeclareTextSymbol{\textthyphenchar}{OT4}{'\-}
783 %\DeclareTextSymbol{\textthyphen}{OT4}{'\-}
784 \DeclareTextSymbol{\textquestiondown}{OT4}{62}
785 \DeclareTextSymbol{\textquotedblleft}{OT4}{92}
786 \DeclareTextSymbol{\textquotedblright}{OT4}{'\'}
787 \DeclareTextSymbol{\textquoteleft}{OT4}{'\'}
788 \DeclareTextSymbol{\textquoteright}{OT4}{'\'}

```

Definition for Å as in OT1:

```

789 \DeclareTextCompositeCommand{\r}{OT4}{A}
790   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
791    \rlap{\raise.67\dimen@\hbox{\char23}}A}

```

In the OT4 encoding, £ and \$ share a slot.

```

792 \DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
793   \ifdim \fontdimen\@ne\font >\z@
794     \slshape
795   \else
796     \upshape
797   \fi
798   \char'\$ \egroup}
799 \DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
800   \ifdim \fontdimen\@ne\font >\z@
801     \itshape
802   \else
803     \fontshape{ui}\selectfont
804   \fi
805   \char'\$ \egroup}

```

Declare the composites.

```

806 \DeclareTextComposite{\k}{OT4}{A}{129}
807 \DeclareTextComposite{\'}{OT4}{C}{130}
808 \DeclareTextComposite{\k}{OT4}{E}{134}
809 \DeclareTextComposite{\'}{OT4}{N}{139}
810 \DeclareTextComposite{\'}{OT4}{S}{145}
811 \DeclareTextComposite{\'}{OT4}{Z}{153}
812 \DeclareTextComposite{\.}{OT4}{Z}{155}
813 \DeclareTextComposite{\k}{OT4}{a}{161}
814 \DeclareTextComposite{\'}{OT4}{c}{162}
815 \DeclareTextComposite{\k}{OT4}{e}{166}
816 \DeclareTextComposite{\'}{OT4}{n}{171}
817 \DeclareTextComposite{\'}{OT4}{s}{177}
818 \DeclareTextComposite{\'}{OT4}{z}{185}
819 \DeclareTextComposite{\.}{OT4}{z}{187}
820 \DeclareTextComposite{\'}{OT4}{O}{211}
821 \DeclareTextComposite{\'}{OT4}{o}{243}
822 </OT4>

```

20.10 Definitions for the TS1 encoding

```

823 <*TS1>
824 \DeclareFontEncoding{TS1}{}{}
825 \DeclareFontSubstitution{TS1}{cmr}{m}{n}

```

Some accents have to be built by hand. Note that \ooalign and \o@lign must be inside a group.

```

826 \DeclareTextCommand{\capitalcedilla}{TS1}[1]
827   {\hmode@bgroup
828    \ooalign{\null#1\crrc\hidewidth\char11\hidewidth}\egroup}
829 \DeclareTextCommand{\capitalogonek}{TS1}[1]
830   {\hmode@bgroup
831    \ooalign{\null#1\crrc\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

```
832 \DeclareTextAccent{\capitalgrave}{TS1}{0}
833 \DeclareTextAccent{\capitalacute}{TS1}{1}
834 \DeclareTextAccent{\capitalcircumflex}{TS1}{2}
835 \DeclareTextAccent{\capitaltilde}{TS1}{3}
836 \DeclareTextAccent{\capitaldieresis}{TS1}{4}
837 \DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}
838 \DeclareTextAccent{\capitalring}{TS1}{6}
839 \DeclareTextAccent{\capitalcaron}{TS1}{7}
```

"08 = 8

```
840 \DeclareTextAccent{\capitalbreve}{TS1}{8}
841 \DeclareTextAccent{\capitalmacron}{TS1}{9}
842 \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.

" =

```
843 \DeclareTextAccent{\t}{TS1}{26}
844 \DeclareTextAccent{\capitaltie}{TS1}{27}
845 \DeclareTextAccent{\newtie}{TS1}{28}
846 \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`.

```
847 \DeclareTextSymbol{\textcapitalcompwordmark}{TS1}{23}
848 \DeclareTextSymbol{\textascendercompwordmark}{TS1}{31}
```

The text companion symbols.

```
849 \DeclareTextSymbol{\textquotestraightbase}{TS1}{13}

"10 = 16

850 \DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
851 \DeclareTextSymbol{\texttwelveudash}{TS1}{21}
852 \DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}

"18 = 24

853 \DeclareTextSymbol{\textleftarrow}{TS1}{24}
854 \DeclareTextSymbol{\textrightarrow}{TS1}{25}

"20 = 32

855 \DeclareTextSymbol{\textblank}{TS1}{32}
856 \DeclareTextSymbol{\textdollar}{TS1}{36}
857 \DeclareTextSymbol{\textquotesingle}{TS1}{39}
```

"28 = 40

```
858 \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.

```
859 \DeclareTextSymbol{\textdblhyphen}{TS1}{45}
```

```
860 \DeclareTextSymbol{\textfractionsolidus}{TS1}{47}
```

Oldstyle digits.

"30 = 48

```
861 \DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
```

```
862 \DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
```

```
863 \DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
```

```
864 \DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
```

```
865 \DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
```

```
866 \DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
```

```
867 \DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
```

```
868 \DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}
```

"38 = 56

```
869 \DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
```

```
870 \DeclareTextSymbol{\textnineoldstyle}{TS1}{57}
```

More text companion symbols.

```
871 \DeclareTextSymbol{\textlangle}{TS1}{60}
```

```
872 \DeclareTextSymbol{\textminus}{TS1}{61}
```

```
873 \DeclareTextSymbol{\textrangle}{TS1}{62}
```

"48 = 72

```
874 \DeclareTextSymbol{\textmho}{TS1}{77}
```

The big circle is here to define the command `\textcircled`. Formerly it was taken from the `cmsy` font.

```
875 \DeclareTextSymbol{\textbigcircle}{TS1}{79}
```

```
876 \DeclareTextCommand{\textcircled}{TS1}[1]{\hmode\bgroup
```

```
877   \oalign{%
```

```
878     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
```

```
879     \char 79   % '117 = "4F
```

```
880   }%
```

```
881 \egroup}
```

More text companion symbols.

"50 = 80

```
882 \DeclareTextSymbol{\textohm}{TS1}{87}
```

"58 = 88

```
883 \DeclareTextSymbol{\textlbrackdbl}{TS1}{91}
```

```
884 \DeclareTextSymbol{\textrbrackdbl}{TS1}{93}
```

```
885 \DeclareTextSymbol{\textuparrow}{TS1}{94}
```

```
886 \DeclareTextSymbol{\textdownarrow}{TS1}{95}
```

"60 = 96

```
887 \DeclareTextSymbol{\textasciigrave}{TS1}{96}
```

```
888 \DeclareTextSymbol{\textborn}{TS1}{98}
```

```
889 \DeclareTextSymbol{\textdivorced}{TS1}{99}
```

```
890 \DeclareTextSymbol{\textdied}{TS1}{100}
```

”68 = 104

```
891 \DeclareTextSymbol{\textleaf}{TS1}{108}
892 \DeclareTextSymbol{\textmarried}{TS1}{109}
893 \DeclareTextSymbol{\textmusicalnote}{TS1}{110}
```

”78 = 120

```
894 \DeclareTextSymbol{\texttildebelow}{TS1}{126}
```

This glyph, `\textdblhyphenchar` is hanging, like the hyphenchar of the ec fonts.

```
895 \DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}
```

”80 = 128

```
896 \DeclareTextSymbol{\textasciibreve}{TS1}{128}
897 \DeclareTextSymbol{\textasciicaron}{TS1}{129}
```

This next glyph is *not* the same as `\textquotedbl`.

```
898 \DeclareTextSymbol{\textacutedbl}{TS1}{130}
899 \DeclareTextSymbol{\textgravedbl}{TS1}{131}
900 \DeclareTextSymbol{\textdagger}{TS1}{132}
901 \DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
902 \DeclareTextSymbol{\textbardbl}{TS1}{134}
903 \DeclareTextSymbol{\textperthousand}{TS1}{135}
```

”88 = 136

```
904 \DeclareTextSymbol{\textbullet}{TS1}{136}
905 \DeclareTextSymbol{\textcelsius}{TS1}{137}
906 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
907 \DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
908 \DeclareTextSymbol{\textflorin}{TS1}{140}
909 \DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
910 \DeclareTextSymbol{\textwon}{TS1}{142}
911 \DeclareTextSymbol{\textnaira}{TS1}{143}
```

”90 = 144

```
912 \DeclareTextSymbol{\textguarani}{TS1}{144}
913 \DeclareTextSymbol{\textpeso}{TS1}{145}
914 \DeclareTextSymbol{\textlira}{TS1}{146}
915 \DeclareTextSymbol{\textrecipe}{TS1}{147}
916 \DeclareTextSymbol{\textinterrobang}{TS1}{148}
917 \DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
918 \DeclareTextSymbol{\textdong}{TS1}{150}
919 \DeclareTextSymbol{\texttrademark}{TS1}{151}
```

”98 = 152

```
920 \DeclareTextSymbol{\textpertenthousand}{TS1}{152}
921 \DeclareTextSymbol{\textpilcrow}{TS1}{153}
922 \DeclareTextSymbol{\textbaht}{TS1}{154}
923 \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`.

```
924 \DeclareTextSymbol{\textdiscount}{TS1}{156}
925 \DeclareTextSymbol{\textestimated}{TS1}{157}
926 \DeclareTextSymbol{\textopenbullet}{TS1}{158}
927 \DeclareTextSymbol{\textservicemark}{TS1}{159}
```

"A0 = 160

```
928 \DeclareTextSymbol{\textlquill}{TS1}{160}
929 \DeclareTextSymbol{\textrquill}{TS1}{161}
930 \DeclareTextSymbol{\textcent}{TS1}{162}
931 \DeclareTextSymbol{\textsterling}{TS1}{163}
932 \DeclareTextSymbol{\textcurrency}{TS1}{164}
933 \DeclareTextSymbol{\textyen}{TS1}{165}
934 \DeclareTextSymbol{\textbrokenbar}{TS1}{166}
935 \DeclareTextSymbol{\textsection}{TS1}{167}
```

"A8 = 168

```
936 \DeclareTextSymbol{\textasciidieresis}{TS1}{168}
937 \DeclareTextSymbol{\textcopyright}{TS1}{169}
938 \DeclareTextSymbol{\textordfeminine}{TS1}{170}
939 \DeclareTextSymbol{\textcopyleft}{TS1}{171}
940 \DeclareTextSymbol{\textlnot}{TS1}{172}
```

The meaning of the circled-P is "sound recording copyright".

```
941 \DeclareTextSymbol{\textcircledP}{TS1}{173}
942 \DeclareTextSymbol{\textregistered}{TS1}{174}
943 \DeclareTextSymbol{\textasciimacron}{TS1}{175}
```

"B0 = 176

```
944 \DeclareTextSymbol{\textdegree}{TS1}{176}
945 \DeclareTextSymbol{\textpm}{TS1}{177}
946 \DeclareTextSymbol{\texttwosuperior}{TS1}{178}
947 \DeclareTextSymbol{\textthreesuperior}{TS1}{179}
948 \DeclareTextSymbol{\textasciicute}{TS1}{180}
949 \DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
950 \DeclareTextSymbol{\textparagraph}{TS1}{182}
951 \DeclareTextSymbol{\textperiodcentered}{TS1}{183}
```

"B8 = 184

```
952 \DeclareTextSymbol{\textreferencemark}{TS1}{184}
953 \DeclareTextSymbol{\textonesuperior}{TS1}{185}
954 \DeclareTextSymbol{\textordmasculine}{TS1}{186}
955 \DeclareTextSymbol{\textsurd}{TS1}{187}
956 \DeclareTextSymbol{\textonequarter}{TS1}{188}
957 \DeclareTextSymbol{\textonehalf}{TS1}{189}
958 \DeclareTextSymbol{\textthreequarters}{TS1}{190}
959 \DeclareTextSymbol{\texteuro}{TS1}{191}
```

"E0 = 208

```
960 \DeclareTextSymbol{\texttimes}{TS1}{214}
```

"F0 = 240

```
961 \DeclareTextSymbol{\textdiv}{TS1}{246}
962 \</TS1>
```

20.11 Definitions for the TU encoding

The TU encoding was originally introduced in the contributed package `fontspec` as a Unicode encoding for XeTeX and LuaTeX.

Normally for these engines, the input consists of Unicode characters encoded in UTF-8. There is therefore little need to use the traditional (ASCII) encoding-specific commands

However, sometimes (e.g. for backwards compatibility) it can be useful to access these Unicode characters via such ASCII-based markup. The commands provided here cover the characters in the T1 and TS1 encodings, but specified in Unicode position. Almost all the command names have been mechanically extracted from the `inputenc` UTF-8 support, which is essentially doing a reverse mapping from UTF-8 data to L^AT_EX LICR commands.

A few additional names for character which were supported in the original `fontspec` version of this file have also been added, even though they are not currently in the default `inputenc` UTF-8 declarations.

963 `{*TU}`

In the base interface the Unicode encoding is always known as TU. But we parameterise the encoding name to allow for modelling differences in Unicode support by different fonts.

964 `\providecommand\UnicodeEncodingName{TU}`

As the Unicode encoding, TU, is only currently available with XeTeX or LuaTeX, we detect these engines first, and make adjustments for the differing font loading syntax. For other engines, we issue a warning then abort this file, switching back to T1 encoding.

965 `\begingroup\expandafter\expandafter\expandafter\endgroup`

966 `\expandafter\ifx\csname XeTeXrevision\endcsname\relax`

967 `\begingroup\expandafter\expandafter\expandafter\endgroup`

968 `\expandafter\ifx\csname directlua\endcsname\relax`

Not LuaTeX or XeTeX, abort with a warning.

969 `\PackageWarningNoLine{fontenc}`

970 `{\UnicodeEncodingName\space`

971 `encoding is only available with XeTeX and LuaTeX.\MessageBreak`

972 `Defaulting to T1 encoding}`

973 `\def\encodingdefault{T1}`

974 `\expandafter\expandafter\expandafter\endinput`

975 `\else`

LuaTeX.

976 `\def\UnicodeFontTeXLigatures{+tlig;}`

977 `\def\reserved@a#1{%`

978 `\def\@remove@tlig##1{\@remove@tlig@##1\@nil#1\@nil\relax}`

979 `\def\@remove@tlig@##1#1{\@remove@tlig@##1}}`

980 `\edef\reserved@b{\detokenize{+tlig;}}`

981 `\expandafter\reserved@a\expandafter{\reserved@b}`

982 `\def\@remove@tlig@@#1\@nil#2\relax{#1}`

983 `\def\remove@tlig#1{%`

984 `\begingroup`

985 `\font\remove@tlig`

986 `\expandafter\@remove@tlig\expandafter{\fontname\font}%`

987 `\remove@tlig`

988 `\char#1\relax`

989 `\endgroup`

990 `}`

991 `\fi`

992 `\else`

XeTeX

```
993 \def\UnicodeFontTeXLigatures{mapping=tex-text;}
994 \def\remove@tlig#1{\XeTeXglyph\numexpr\XeTeXcharglyph#1\relax}
995 \fi
996 \def\UnicodeFontFile#1#2{"[#1]:#2"}
997 \def\UnicodeFontName#1#2{"#1:#2"}

Declare the encoding
998 \DeclareFontEncoding\UnicodeEncodingName{}{}

Declare accent command to use a postpended combining character rather than
the TeX \accent primitive
999 \def\add@unicode@accent#1#2{%
1000 \if\relax\detokenize{#2}\relax~^a0\else#2\fi
1001 \char#1\relax}
1002 \def\DeclareUnicodeAccent#1#2#3{%
1003 \DeclareTextCommand{#1}{#2}{\add@unicode@accent{#3}}%
1004 }

Wrapper around \DeclareTextCompositeCommand that uses the declared com-
posite if it exists in the current font or falls back to the default definition for the
TU accent if not.
1005 {
1006 \catcode\z@=11\relax
1007 \gdef\DeclareUnicodeComposite#1#2#3{%
1008 \def\reserved@a##1##2{%
1009 \DeclareTextCompositeCommand#1\UnicodeEncodingName{#2}{%
1010 \iffontchar\font#3 ##2%
1011 \else ##1\fi}}%
1012 \expandafter\expandafter\expandafter\extract@default@composite
1013 \csname\UnicodeEncodingName\string#1\endcsname{#2}\@nil
1014 \bgroup
1015 \lccode\z@#3 %
1016 \lowercase{\egroup
1017 \expandafter\reserved@a\expandafter{\reserved@b}{~^@}}}%
1018 }

1019 \def\extract@default@composite#1{%
1020 \ifx\@text@composite#1%
1021 \expandafter\extract@default@composite@a
1022 \else
1023 \expandafter\extract@default@composite@b\expandafter#1%
1024 \fi}

1025 \def\extract@default@composite@a#1\@text@composite#2\@nil{%
1026 \def\reserved@b{#2}}
1027 \def\extract@default@composite@b#1#2\@nil{%
1028 \def\reserved@b{#1#2}}

1029 \DeclareTextCommand\textquotesingle \UnicodeEncodingName{%
1030 \remove@tlig{"0027}}
1031 \DeclareTextCommand\textasciigrave \UnicodeEncodingName{%
1032 \remove@tlig{"0060}}
1033 \DeclareTextCommand\textquotedbl \UnicodeEncodingName{%
1034 \remove@tlig{"0022}}
```

1035 \DeclareTextSymbol{\textdollar}	\UnicodeEncodingName{"0024}
1036 \DeclareTextSymbol{\textless}	\UnicodeEncodingName{"003C}
1037 \DeclareTextSymbol{\textgreater}	\UnicodeEncodingName{"003E}
1038 \DeclareTextSymbol{\textbackslash}	\UnicodeEncodingName{"005C}
1039 \DeclareTextSymbol{\textasciicircum}	\UnicodeEncodingName{"005E}
1040 \DeclareTextSymbol{\textunderscore}	\UnicodeEncodingName{"005F}
1041 \DeclareTextSymbol{\textbraceleft}	\UnicodeEncodingName{"007B}
1042 \DeclareTextSymbol{\textbar}	\UnicodeEncodingName{"007C}
1043 \DeclareTextSymbol{\textbraceright}	\UnicodeEncodingName{"007D}
1044 \DeclareTextSymbol{\textasciitilde}	\UnicodeEncodingName{"007E}
1045 \DeclareTextSymbol{\textexclamdown}	\UnicodeEncodingName{"00A1}
1046 \DeclareTextSymbol{\textcent}	\UnicodeEncodingName{"00A2}
1047 \DeclareTextSymbol{\textsterling}	\UnicodeEncodingName{"00A3}
1048 \DeclareTextSymbol{\textcurrency}	\UnicodeEncodingName{"00A4}
1049 \DeclareTextSymbol{\textyen}	\UnicodeEncodingName{"00A5}
1050 \DeclareTextSymbol{\textbrokenbar}	\UnicodeEncodingName{"00A6}
1051 \DeclareTextSymbol{\textsection}	\UnicodeEncodingName{"00A7}
1052 \DeclareTextSymbol{\textasciidieresis}	\UnicodeEncodingName{"00A8}
1053 \DeclareTextSymbol{\textcopyright}	\UnicodeEncodingName{"00A9}
1054 \DeclareTextSymbol{\textordfeminine}	\UnicodeEncodingName{"00AA}
1055 \DeclareTextSymbol{\guillemetleft}	\UnicodeEncodingName{"00AB}
1056 % old Adobe name	
1057 \DeclareTextSymbol{\guillemotleft}	\UnicodeEncodingName{"00AB}
1058 \DeclareTextSymbol{\textlnot}	\UnicodeEncodingName{"00AC}
1059 \DeclareTextSymbol{\textregistered}	\UnicodeEncodingName{"00AE}
1060 \DeclareTextSymbol{\textasciimacron}	\UnicodeEncodingName{"00AF}
1061 \DeclareTextSymbol{\textdegree}	\UnicodeEncodingName{"00B0}
1062 \DeclareTextSymbol{\textpm}	\UnicodeEncodingName{"00B1}
1063 \DeclareTextSymbol{\texttwosuperior}	\UnicodeEncodingName{"00B2}
1064 \DeclareTextSymbol{\textthreesuperior}	\UnicodeEncodingName{"00B3}
1065 \DeclareTextSymbol{\textasciacute}	\UnicodeEncodingName{"00B4}
1066 \DeclareTextSymbol{\textmu}	\UnicodeEncodingName{"00B5}
1067 \DeclareTextSymbol{\textparagraph}	\UnicodeEncodingName{"00B6}
1068 \DeclareTextSymbol{\textperiodcentered}	\UnicodeEncodingName{"00B7}
1069 \DeclareTextSymbol{\textonesuperior}	\UnicodeEncodingName{"00B9}
1070 \DeclareTextSymbol{\textordmasculine}	\UnicodeEncodingName{"00BA}
1071 \DeclareTextSymbol{\guillemetright}	\UnicodeEncodingName{"00BB}
1072 % old Adobe name	
1073 \DeclareTextSymbol{\guillemotright}	\UnicodeEncodingName{"00BB}
1074 \DeclareTextSymbol{\textonequarter}	\UnicodeEncodingName{"00BC}
1075 \DeclareTextSymbol{\textonehalf}	\UnicodeEncodingName{"00BD}
1076 \DeclareTextSymbol{\textthreequarters}	\UnicodeEncodingName{"00BE}
1077 \DeclareTextSymbol{\textquestiondown}	\UnicodeEncodingName{"00BF}
1078 \DeclareTextSymbol{\AE}	\UnicodeEncodingName{"00C6}
1079 \DeclareTextSymbol{\DH}	\UnicodeEncodingName{"00D0}
1080 \DeclareTextSymbol{\textttimes}	\UnicodeEncodingName{"00D7}
1081 \DeclareTextSymbol{\O}	\UnicodeEncodingName{"00D8}
1082 \DeclareTextSymbol{\TH}	\UnicodeEncodingName{"00DE}
1083 \DeclareTextSymbol{\ss}	\UnicodeEncodingName{"00DF}
1084 \DeclareTextSymbol{\ae}	\UnicodeEncodingName{"00E6}
1085 \DeclareTextSymbol{\dh}	\UnicodeEncodingName{"00F0}
1086 \DeclareTextSymbol{\textdiv}	\UnicodeEncodingName{"00F7}

1087 \DeclareTextSymbol{\o}	\UnicodeEncodingName{"00F8}
1088 \DeclareTextSymbol{\th}	\UnicodeEncodingName{"00FE}
1089 \DeclareTextSymbol{\DJ}	\UnicodeEncodingName{"0110}
1090 \DeclareTextSymbol{\dj}	\UnicodeEncodingName{"0111}
1091 \DeclareTextSymbol{\i}	\UnicodeEncodingName{"0131}
1092 \DeclareTextSymbol{\IJ}	\UnicodeEncodingName{"0132}
1093 \DeclareTextSymbol{\ij}	\UnicodeEncodingName{"0133}
1094 \DeclareTextSymbol{\L}	\UnicodeEncodingName{"0141}
1095 \DeclareTextSymbol{\l}	\UnicodeEncodingName{"0142}
1096 \DeclareTextSymbol{\NG}	\UnicodeEncodingName{"014A}
1097 \DeclareTextSymbol{\ng}	\UnicodeEncodingName{"014B}
1098 \DeclareTextSymbol{\OE}	\UnicodeEncodingName{"0152}
1099 \DeclareTextSymbol{\oe}	\UnicodeEncodingName{"0153}
1100 \DeclareTextSymbol{\textflorin}	\UnicodeEncodingName{"0192}
1101 \DeclareTextSymbol{\j}	\UnicodeEncodingName{"0237}
1102 \DeclareTextSymbol{\textasciicaron}	\UnicodeEncodingName{"02C7}
1103 \DeclareTextSymbol{\textasciibreve}	\UnicodeEncodingName{"02D8}
1104 \DeclareTextSymbol{\textacutdbl}	\UnicodeEncodingName{"02DD}
1105 \DeclareTextSymbol{\textgravedbl}	\UnicodeEncodingName{"02F5}
1106 \DeclareTextSymbol{\texttildebelow}	\UnicodeEncodingName{"02F7}
1107 \DeclareTextSymbol{\textbaht}	\UnicodeEncodingName{"0E3F}
1108 \DeclareTextSymbol{\SS}	\UnicodeEncodingName{"1E9E}
1109 \DeclareTextSymbol{\textcompwordmark}	\UnicodeEncodingName{"200C}
1110 \DeclareTextSymbol{\textendash}	\UnicodeEncodingName{"2013}
1111 \DeclareTextSymbol{\textemdash}	\UnicodeEncodingName{"2014}
1112 \DeclareTextSymbol{\textbardbl}	\UnicodeEncodingName{"2016}
1113 \DeclareTextSymbol{\textquoteleft}	\UnicodeEncodingName{"2018}
1114 \DeclareTextSymbol{\textquoteright}	\UnicodeEncodingName{"2019}
1115 \DeclareTextSymbol{\quotesinglbase}	\UnicodeEncodingName{"201A}
1116 \DeclareTextSymbol{\textquotedblleft}	\UnicodeEncodingName{"201C}
1117 \DeclareTextSymbol{\textquotedblright}	\UnicodeEncodingName{"201D}
1118 \DeclareTextSymbol{\quotedblbase}	\UnicodeEncodingName{"201E}
1119 \DeclareTextSymbol{\textdagger}	\UnicodeEncodingName{"2020}
1120 \DeclareTextSymbol{\textdaggerdbl}	\UnicodeEncodingName{"2021}
1121 \DeclareTextSymbol{\textbullet}	\UnicodeEncodingName{"2022}
1122 \DeclareTextSymbol{\textellipsis}	\UnicodeEncodingName{"2026}
1123 \DeclareTextSymbol{\textperthousand}	\UnicodeEncodingName{"2030}
1124 \DeclareTextSymbol{\textpertenthousand}	\UnicodeEncodingName{"2031}
1125 \DeclareTextSymbol{\guilsinglleft}	\UnicodeEncodingName{"2039}
1126 \DeclareTextSymbol{\guilsinglright}	\UnicodeEncodingName{"203A}
1127 \DeclareTextSymbol{\textreferencemark}	\UnicodeEncodingName{"203B}
1128 \DeclareTextSymbol{\textinterrobang}	\UnicodeEncodingName{"203D}
1129 \DeclareTextSymbol{\textfractionsolidus}	\UnicodeEncodingName{"2044}
1130 \DeclareTextSymbol{\textlquill}	\UnicodeEncodingName{"2045}
1131 \DeclareTextSymbol{\textrquill}	\UnicodeEncodingName{"2046}
1132 \DeclareTextSymbol{\textdiscount}	\UnicodeEncodingName{"2052}
1133 \DeclareTextSymbol{\textcolonmonetary}	\UnicodeEncodingName{"20A1}
1134 \DeclareTextSymbol{\textlira}	\UnicodeEncodingName{"20A4}
1135 \DeclareTextSymbol{\textnaira}	\UnicodeEncodingName{"20A6}
1136 \DeclareTextSymbol{\textwon}	\UnicodeEncodingName{"20A9}
1137 \DeclareTextSymbol{\textdong}	\UnicodeEncodingName{"20AB}
1138 \DeclareTextSymbol{\texteuro}	\UnicodeEncodingName{"20AC}
1139 \DeclareTextSymbol{\textpeso}	\UnicodeEncodingName{"20B1}
1140 \DeclareTextSymbol{\textcelsius}	\UnicodeEncodingName{"2103}

```

1141 \DeclareTextSymbol{\textnumero} \UnicodeEncodingName{"2116}
1142 \DeclareTextSymbol{\textcircledP} \UnicodeEncodingName{"2117}
1143 \DeclareTextSymbol{\textrecipe} \UnicodeEncodingName{"211E}
1144 \DeclareTextSymbol{\textservice} \UnicodeEncodingName{"2120}
1145 \DeclareTextSymbol{\texttrademark} \UnicodeEncodingName{"2122}
1146 \DeclareTextSymbol{\textohm} \UnicodeEncodingName{"2126}
1147 \DeclareTextSymbol{\textmho} \UnicodeEncodingName{"2127}
1148 \DeclareTextSymbol{\textestimated} \UnicodeEncodingName{"212E}
1149 \DeclareTextSymbol{\textleftarrow} \UnicodeEncodingName{"2190}
1150 \DeclareTextSymbol{\textuparrow} \UnicodeEncodingName{"2191}
1151 \DeclareTextSymbol{\textrightarrow} \UnicodeEncodingName{"2192}
1152 \DeclareTextSymbol{\textdownarrow} \UnicodeEncodingName{"2193}
1153 \DeclareTextSymbol{\textminus} \UnicodeEncodingName{"2212}
1154
1155 \DeclareTextSymbol{\Hwithstroke} \UnicodeEncodingName{"0126}
1156 \DeclareTextSymbol{\hwithstroke} \UnicodeEncodingName{"0127}

```

Not all fonts have U+2217 but using U+002A requires some adjustment.

```

1157 \DeclareTextCommand{\textasteriskcentered}\UnicodeEncodingName{%
1158   \iffontchar\font"2217 \char"2217 \else
1159     \begingroup
1160       \fontsize
1161       {\the\dimexpr1.2\dimexpr\f@size pt\relax}%
1162       {\f@baselineskip}%
1163       \selectfont
1164       \raisebox{-0.6ex}{\dimexpr\height-0.6ex}[0pt]{*}%
1165     \endgroup
1166   \fi
1167 }
1168 \DeclareTextSymbol{\textsurd} \UnicodeEncodingName{"221A}
1169 \DeclareTextSymbol{\textlangle} \UnicodeEncodingName{"2329}
1170 \DeclareTextSymbol{\textrangle} \UnicodeEncodingName{"232A}
1171 \DeclareTextSymbol{\textblank} \UnicodeEncodingName{"2422}
1172 \DeclareTextSymbol{\textvisiblespace} \UnicodeEncodingName{"2423}
1173 \DeclareTextSymbol{\textopenbullet} \UnicodeEncodingName{"25E6}
1174 \DeclareTextSymbol{\textbigcircle} \UnicodeEncodingName{"25EF}
1175 \DeclareTextSymbol{\textmusicalnote} \UnicodeEncodingName{"266A}
1176 \DeclareTextSymbol{\textmarried} \UnicodeEncodingName{"26AD}
1177 \DeclareTextSymbol{\textdivorced} \UnicodeEncodingName{"26AE}
1178 \DeclareTextSymbol{\textinterrobangdown} \UnicodeEncodingName{"2E18}

```

Accents must be declared before the composites that use them.

```

1179 \DeclareUnicodeAccent{\` } \UnicodeEncodingName{"0300}
1180 \DeclareUnicodeAccent{\' } \UnicodeEncodingName{"0301}
1181 \DeclareUnicodeAccent{\^ } \UnicodeEncodingName{"0302}
1182 \DeclareUnicodeAccent{\~ } \UnicodeEncodingName{"0303}
1183 \DeclareUnicodeAccent{\= } \UnicodeEncodingName{"0304}
1184 \DeclareUnicodeAccent{\u } \UnicodeEncodingName{"0306}
1185 \DeclareUnicodeAccent{\. } \UnicodeEncodingName{"0307}
1186 \DeclareUnicodeAccent{\" } \UnicodeEncodingName{"0308}
1187 \DeclareUnicodeAccent{\r } \UnicodeEncodingName{"030A}
1188 \DeclareUnicodeAccent{\H } \UnicodeEncodingName{"030B}
1189 \DeclareUnicodeAccent{\v } \UnicodeEncodingName{"030C}

```

```

1190 \DeclareUnicodeAccent{\b}          \UnicodeEncodingName{"0332}
1191 \DeclareUnicodeAccent{\d}          \UnicodeEncodingName{"0323}
1192 \DeclareUnicodeAccent{\c}          \UnicodeEncodingName{"0327}
1193 \DeclareUnicodeAccent{\k}          \UnicodeEncodingName{"0328}
1194 \DeclareTextCommand\textcommabelow \UnicodeEncodingName[1]
1195   {\hmode\bgroup\oalign{\null#1\crrc\hidewidth\raise-.31ex
1196     \hbox{\check@mathfonts\fontsize\ssf@size\z@
1197       \math@fontsfalse\selectfont,}\hidewidth}\egroup}

1198 \DeclareUnicodeComposite{\^}        {}{"005E}
1199 \DeclareUnicodeComposite{\~}        {}{"007E}

1200 \DeclareUnicodeComposite{\'}       {A}{"00C0}
1201 \DeclareUnicodeComposite{\'}       {A}{"00C1}
1202 \DeclareUnicodeComposite{\^}       {A}{"00C2}
1203 \DeclareUnicodeComposite{\^}       {A}{"00C3}
1204 \DeclareUnicodeComposite{\"}       {A}{"00C4}
1205 \DeclareUnicodeComposite{\r}       {A}{"00C5}
1206 \DeclareUnicodeComposite{\c}       {C}{"00C7}
1207 \DeclareUnicodeComposite{\'}       {E}{"00C8}
1208 \DeclareUnicodeComposite{\'}       {E}{"00C9}
1209 \DeclareUnicodeComposite{\^}       {E}{"00CA}
1210 \DeclareUnicodeComposite{\"}       {E}{"00CB}
1211 \DeclareUnicodeComposite{\'}       {I}{"00CC}
1212 \DeclareUnicodeComposite{\'}       {I}{"00CD}
1213 \DeclareUnicodeComposite{\^}       {I}{"00CE}
1214 \DeclareUnicodeComposite{\"}       {I}{"00CF}
1215 \DeclareUnicodeComposite{\^}       {N}{"00D1}
1216 \DeclareUnicodeComposite{\'}       {O}{"00D2}
1217 \DeclareUnicodeComposite{\'}       {O}{"00D3}
1218 \DeclareUnicodeComposite{\^}       {O}{"00D4}
1219 \DeclareUnicodeComposite{\^}       {O}{"00D5}
1220 \DeclareUnicodeComposite{\"}       {O}{"00D6}
1221 \DeclareUnicodeComposite{\'}       {U}{"00D9}
1222 \DeclareUnicodeComposite{\'}       {U}{"00DA}
1223 \DeclareUnicodeComposite{\^}       {U}{"00DB}
1224 \DeclareUnicodeComposite{\"}       {U}{"00DC}
1225 \DeclareUnicodeComposite{\'}       {Y}{"00DD}
1226 \DeclareUnicodeComposite{\'}       {a}{"00E0}
1227 \DeclareUnicodeComposite{\'}       {a}{"00E1}
1228 \DeclareUnicodeComposite{\^}       {a}{"00E2}
1229 \DeclareUnicodeComposite{\^}       {a}{"00E3}
1230 \DeclareUnicodeComposite{\"}       {a}{"00E4}
1231 \DeclareUnicodeComposite{\r}       {a}{"00E5}
1232 \DeclareUnicodeComposite{\c}       {c}{"00E7}
1233 \DeclareUnicodeComposite{\'}       {e}{"00E8}
1234 \DeclareUnicodeComposite{\'}       {e}{"00E9}
1235 \DeclareUnicodeComposite{\^}       {e}{"00EA}
1236 \DeclareUnicodeComposite{\"}       {e}{"00EB}
1237 \DeclareUnicodeComposite{\'}       \i {"00EC}
1238 \DeclareUnicodeComposite{\'}       {i}{"00EC}
1239 \DeclareUnicodeComposite{\'}       \i {"00ED}
1240 \DeclareUnicodeComposite{\'}       {i}{"00ED}
1241 \DeclareUnicodeComposite{\^}       \i {"00EE}
1242 \DeclareUnicodeComposite{\^}       {i}{"00EE}

```

1243 \DeclareUnicodeComposite{"}	\i {"00EF}
1244 \DeclareUnicodeComposite{"}	{i}{"00EF}
1245 \DeclareUnicodeComposite{~}	{n}{"00F1}
1246 \DeclareUnicodeComposite{'}	{o}{"00F2}
1247 \DeclareUnicodeComposite{'}	{o}{"00F3}
1248 \DeclareUnicodeComposite{^}	{o}{"00F4}
1249 \DeclareUnicodeComposite{~}	{o}{"00F5}
1250 \DeclareUnicodeComposite{"}	{o}{"00F6}
1251 \DeclareUnicodeComposite{'}	{u}{"00F9}
1252 \DeclareUnicodeComposite{'}	{u}{"00FA}
1253 \DeclareUnicodeComposite{^}	{u}{"00FB}
1254 \DeclareUnicodeComposite{"}	{u}{"00FC}
1255 \DeclareUnicodeComposite{'}	{y}{"00FD}
1256 \DeclareUnicodeComposite{"}	{y}{"00FF}
1257 \DeclareUnicodeComposite{=}	{A}{"0100}
1258 \DeclareUnicodeComposite{=}	{a}{"0101}
1259 \DeclareUnicodeComposite{u}	{A}{"0102}
1260 \DeclareUnicodeComposite{u}	{a}{"0103}
1261 \DeclareUnicodeComposite{k}	{A}{"0104}
1262 \DeclareUnicodeComposite{k}	{a}{"0105}
1263 \DeclareUnicodeComposite{'}	{C}{"0106}
1264 \DeclareUnicodeComposite{'}	{c}{"0107}
1265 \DeclareUnicodeComposite{^}	{C}{"0108}
1266 \DeclareUnicodeComposite{^}	{c}{"0109}
1267 \DeclareUnicodeComposite{.}	{C}{"010A}
1268 \DeclareUnicodeComposite{.}	{c}{"010B}
1269 \DeclareUnicodeComposite{v}	{C}{"010C}
1270 \DeclareUnicodeComposite{v}	{c}{"010D}
1271 \DeclareUnicodeComposite{v}	{D}{"010E}
1272 \DeclareUnicodeComposite{v}	{d}{"010F}
1273 \DeclareUnicodeComposite{=}	{E}{"0112}
1274 \DeclareUnicodeComposite{=}	{e}{"0113}
1275 \DeclareUnicodeComposite{u}	{E}{"0114}
1276 \DeclareUnicodeComposite{u}	{e}{"0115}
1277 \DeclareUnicodeComposite{.}	{E}{"0116}
1278 \DeclareUnicodeComposite{.}	{e}{"0117}
1279 \DeclareUnicodeComposite{k}	{E}{"0118}
1280 \DeclareUnicodeComposite{k}	{e}{"0119}
1281 \DeclareUnicodeComposite{v}	{E}{"011A}
1282 \DeclareUnicodeComposite{v}	{e}{"011B}
1283 \DeclareUnicodeComposite{^}	{G}{"011C}
1284 \DeclareUnicodeComposite{^}	{g}{"011D}
1285 \DeclareUnicodeComposite{u}	{G}{"011E}
1286 \DeclareUnicodeComposite{u}	{g}{"011F}
1287 \DeclareUnicodeComposite{.}	{G}{"0120}
1288 \DeclareUnicodeComposite{.}	{g}{"0121}
1289 \DeclareUnicodeComposite{c}	{G}{"0122}
1290 \DeclareUnicodeComposite{c}	{g}{"0123}
1291 \DeclareUnicodeComposite{^}	{H}{"0124}
1292 \DeclareUnicodeComposite{^}	{h}{"0125}
1293 \DeclareUnicodeComposite{~}	{I}{"0128}
1294 \DeclareUnicodeComposite{~}	\i {"0129}
1295 \DeclareUnicodeComposite{~}	{i}{"0129}
1296 \DeclareUnicodeComposite{=}	{I}{"012A}

1297 \DeclareUnicodeComposite{=}	\i {"012B}
1298 \DeclareUnicodeComposite{=}	{i}{"012B}
1299 \DeclareUnicodeComposite{u}	{I}{"012C}
1300 \DeclareUnicodeComposite{u}	\i {"012D}
1301 \DeclareUnicodeComposite{u}	{i}{"012D}
1302 \DeclareUnicodeComposite{k}	{I}{"012E}
1303 \DeclareUnicodeComposite{k}	\i {"012F}
1304 \DeclareUnicodeComposite{k}	{i}{"012F}
1305 \DeclareUnicodeComposite{.}	{I}{"0130}
1306 \DeclareUnicodeComposite{^}	{J}{"0134}
1307 \DeclareUnicodeComposite{^}	\j {"0135}
1308 \DeclareUnicodeComposite{^}	{j}{"0135}
1309 \DeclareUnicodeComposite{c}	{K}{"0136}
1310 \DeclareUnicodeComposite{c}	{k}{"0137}
1311 \DeclareUnicodeComposite{'}	{L}{"0139}
1312 \DeclareUnicodeComposite{'}	{l}{"013A}
1313 \DeclareUnicodeComposite{c}	{L}{"013B}
1314 \DeclareUnicodeComposite{c}	{l}{"013C}
1315 \DeclareUnicodeComposite{v}	{L}{"013D}
1316 \DeclareUnicodeComposite{v}	{l}{"013E}
1317 \DeclareUnicodeComposite{'}	{N}{"0143}
1318 \DeclareUnicodeComposite{'}	{n}{"0144}
1319 \DeclareUnicodeComposite{c}	{N}{"0145}
1320 \DeclareUnicodeComposite{c}	{n}{"0146}
1321 \DeclareUnicodeComposite{v}	{N}{"0147}
1322 \DeclareUnicodeComposite{v}	{n}{"0148}
1323 \DeclareUnicodeComposite{=}	{O}{"014C}
1324 \DeclareUnicodeComposite{=}	{o}{"014D}
1325 \DeclareUnicodeComposite{u}	{O}{"014E}
1326 \DeclareUnicodeComposite{u}	{o}{"014F}
1327 \DeclareUnicodeComposite{H}	{O}{"0150}
1328 \DeclareUnicodeComposite{H}	{o}{"0151}
1329 \DeclareUnicodeComposite{'}	{R}{"0154}
1330 \DeclareUnicodeComposite{'}	{r}{"0155}
1331 \DeclareUnicodeComposite{c}	{R}{"0156}
1332 \DeclareUnicodeComposite{c}	{r}{"0157}
1333 \DeclareUnicodeComposite{v}	{R}{"0158}
1334 \DeclareUnicodeComposite{v}	{r}{"0159}
1335 \DeclareUnicodeComposite{'}	{S}{"015A}
1336 \DeclareUnicodeComposite{'}	{s}{"015B}
1337 \DeclareUnicodeComposite{^}	{S}{"015C}
1338 \DeclareUnicodeComposite{^}	{s}{"015D}
1339 \DeclareUnicodeComposite{c}	{S}{"015E}
1340 \DeclareUnicodeComposite{c}	{s}{"015F}
1341 \DeclareUnicodeComposite{v}	{S}{"0160}
1342 \DeclareUnicodeComposite{v}	{s}{"0161}
1343 \DeclareUnicodeComposite{c}	{T}{"0162}
1344 \DeclareUnicodeComposite{c}	{t}{"0163}
1345 \DeclareUnicodeComposite{v}	{T}{"0164}
1346 \DeclareUnicodeComposite{v}	{t}{"0165}
1347 \DeclareUnicodeComposite{~}	{U}{"0168}
1348 \DeclareUnicodeComposite{~}	{u}{"0169}
1349 \DeclareUnicodeComposite{=}	{U}{"016A}
1350 \DeclareUnicodeComposite{=}	{u}{"016B}

1351 \DeclareUnicodeComposite{\u}	{U}{ "016C}
1352 \DeclareUnicodeComposite{\u}	{u}{ "016D}
1353 \DeclareUnicodeComposite{\r}	{U}{ "016E}
1354 \DeclareUnicodeComposite{\r}	{u}{ "016F}
1355 \DeclareUnicodeComposite{\H}	{U}{ "0170}
1356 \DeclareUnicodeComposite{\H}	{u}{ "0171}
1357 \DeclareUnicodeComposite{\k}	{U}{ "0172}
1358 \DeclareUnicodeComposite{\k}	{u}{ "0173}
1359 \DeclareUnicodeComposite{\^}	{W}{ "0174}
1360 \DeclareUnicodeComposite{\^}	{w}{ "0175}
1361 \DeclareUnicodeComposite{\^}	{Y}{ "0176}
1362 \DeclareUnicodeComposite{\^}	{y}{ "0177}
1363 \DeclareUnicodeComposite{"}	{Y}{ "0178}
1364 \DeclareUnicodeComposite{'}	{Z}{ "0179}
1365 \DeclareUnicodeComposite{'}	{z}{ "017A}
1366 \DeclareUnicodeComposite{.}	{Z}{ "017B}
1367 \DeclareUnicodeComposite{.}	{z}{ "017C}
1368 \DeclareUnicodeComposite{\v}	{Z}{ "017D}
1369 \DeclareUnicodeComposite{\v}	{z}{ "017E}
1370 \DeclareUnicodeComposite{\v}	{A}{ "01CD}
1371 \DeclareUnicodeComposite{\v}	{a}{ "01CE}
1372 \DeclareUnicodeComposite{\v}	{I}{ "01CF}
1373 \DeclareUnicodeComposite{\v}	\i { "01D0}
1374 \DeclareUnicodeComposite{\v}	{i}{ "01D0}
1375 \DeclareUnicodeComposite{\v}	{O}{ "01D1}
1376 \DeclareUnicodeComposite{\v}	{o}{ "01D2}
1377 \DeclareUnicodeComposite{\v}	{U}{ "01D3}
1378 \DeclareUnicodeComposite{\v}	{u}{ "01D4}
1379 \DeclareUnicodeComposite{\=}	\AE{ "01E2}
1380 \DeclareUnicodeComposite{\=}	\ae{ "01E3}
1381 \DeclareUnicodeComposite{\v}	{G}{ "01E6}
1382 \DeclareUnicodeComposite{\v}	{g}{ "01E7}
1383 \DeclareUnicodeComposite{\v}	{K}{ "01E8}
1384 \DeclareUnicodeComposite{\v}	{k}{ "01E9}
1385 \DeclareUnicodeComposite{\k}	{O}{ "01EA}
1386 \DeclareUnicodeComposite{\k}	{o}{ "01EB}
1387 \DeclareUnicodeComposite{\v}	\j { "01F0}
1388 \DeclareUnicodeComposite{\v}	{j}{ "01F0}
1389 \DeclareUnicodeComposite{'}	{G}{ "01F4}
1390 \DeclareUnicodeComposite{'}	{g}{ "01F5}
1391 \DeclareUnicodeComposite{\textcommabelow}	{S}{ "0218}
1392 \DeclareUnicodeComposite{\textcommabelow}	{s}{ "0219}
1393 \DeclareUnicodeComposite{\textcommabelow}	{T}{ "021A}
1394 \DeclareUnicodeComposite{\textcommabelow}	{t}{ "021B}
1395 \DeclareUnicodeComposite{\=}	{Y}{ "0232}
1396 \DeclareUnicodeComposite{\=}	{y}{ "0232}
1397 \DeclareUnicodeComposite{.}	{B}{ "1E02}
1398 \DeclareUnicodeComposite{.}	{b}{ "1E03}
1399 \DeclareUnicodeComposite{\d}	{B}{ "1E04}
1400 \DeclareUnicodeComposite{\d}	{b}{ "1E05}
1401 \DeclareUnicodeComposite{\d}	{D}{ "1E0C}
1402 \DeclareUnicodeComposite{\d}	{d}{ "1E0D}
1403 \DeclareUnicodeComposite{\=}	{G}{ "1E20}
1404 \DeclareUnicodeComposite{\=}	{g}{ "1E21}


```

1405 \DeclareUnicodeComposite{\d}      {H}-{ "1E24}
1406 \DeclareUnicodeComposite{\d}      {h}-{ "1E25}
1407 \DeclareUnicodeComposite{\d}      {K}-{ "1E32}
1408 \DeclareUnicodeComposite{\d}      {k}-{ "1E33}
1409 \DeclareUnicodeComposite{\d}      {L}-{ "1E36}
1410 \DeclareUnicodeComposite{\d}      {l}-{ "1E37}
1411 \DeclareUnicodeComposite{\d}      {M}-{ "1E42}
1412 \DeclareUnicodeComposite{\d}      {m}-{ "1E43}
1413 \DeclareUnicodeComposite{\d}      {N}-{ "1E46}
1414 \DeclareUnicodeComposite{\d}      {n}-{ "1E47}
1415 \DeclareUnicodeComposite{\d}      {R}-{ "1E5A}
1416 \DeclareUnicodeComposite{\d}      {r}-{ "1E5B}
1417 \DeclareUnicodeComposite{\d}      {S}-{ "1E62}
1418 \DeclareUnicodeComposite{\d}      {s}-{ "1E63}
1419 \DeclareUnicodeComposite{\d}      {T}-{ "1E6C}
1420 \DeclareUnicodeComposite{\d}      {t}-{ "1E6D}
1421 \DeclareUnicodeComposite{\d}      {V}-{ "1E7E}
1422 \DeclareUnicodeComposite{\d}      {v}-{ "1E7F}
1423 \DeclareUnicodeComposite{\d}      {W}-{ "1E88}
1424 \DeclareUnicodeComposite{\d}      {w}-{ "1E89}
1425 \DeclareUnicodeComposite{\d}      {Z}-{ "1E92}
1426 \DeclareUnicodeComposite{\d}      {z}-{ "1E93}
1427 \DeclareUnicodeComposite{\d}      {A}-{ "1EA0}
1428 \DeclareUnicodeComposite{\d}      {a}-{ "1EA1}
1429 \DeclareUnicodeComposite{\d}      {E}-{ "1EB8}
1430 \DeclareUnicodeComposite{\d}      {e}-{ "1EB9}
1431 \DeclareUnicodeComposite{\d}      {I}-{ "1ECA}
1432 \DeclareUnicodeComposite{\d}      {i}-{ "1ECB}
1433 \DeclareUnicodeComposite{\d}      {O}-{ "1ECC}
1434 \DeclareUnicodeComposite{\d}      {o}-{ "1ECD}
1435 \DeclareUnicodeComposite{\d}      {U}-{ "1EE4}
1436 \DeclareUnicodeComposite{\d}      {u}-{ "1EE5}
1437 \DeclareUnicodeComposite{\d}      {Y}-{ "1EF4}
1438 \DeclareUnicodeComposite{\d}      {y}-{ "1EF5}
1439 </TU>

```

21 Package files

This file now also contains some packages that provide access to the more specialised encodings.

21.1 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`.

In addition the package at the moment contains extra code to extend the `\@uclclist` (list of upper/lower case pairs) for encodings that involve cyrillic characters. THIS IS A TEMPORARY SOLUTION and will not stay this way forever (or so we hope) but right now we are missing a proper interface for this and didn't want to rush it.

1440 (*package)

Here we define a macro that extends the \@uclclist if needed and afterwards turns itself in a noop.

```
1441 \def\update@uclc@with@cyrillic{%
1442   \expandafter\def\expandafter\@uclclist\expandafter
1443   {\@uclclist
1444    \cyr\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
1445    \CYRABHDZE\cyrabhha\CYRABHHA\cyrac\CYRAE\cyrb\CYRB\cyrbyus
1446    \CYRBYUS\cyrc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrds
1447    \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
1448    \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
1449    \CYREPS\cyrerev\CYREREV\cyrry\CYRERY\cyrf\CYRF\cyrfita
1450    \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
1451    \cyrgchcrs\CYRGHCRS\cyrgkh\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
1452    \cyrhdsc\CYRHDSC\cyrrhcrs\CYRRHCRS\cyrrhk\CYRRHK\cyrrhdsn
1453    \CYRRHDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
1454    \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
1455    \cyrkbeak\CYRKBEAK\cyrkds\CYRKDSC\cyrkchcrs\CYRKHCRS\cyrkhk
1456    \CYRKHK\cyrkvcrs\CYRKVCRS\cyr1\CYRL\cyrlldsc\CYRLDSC\cyr1hk
1457    \CYRLHK\cyr1je\CYRLJE\cyrm\CYRM\cyrmdsc\CYRMDSC\cyrmhk\CYRMHK
1458    \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnkh\CYRNHK\cyrnje
1459    \CYRNJE\cyrn1hk\CYRN1HK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
1460    \cyrophk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
1461    \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacr\CYRSACRS
1462    \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
1463    \cyrsftsn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
1464    \cyrt\CYRT\cyrtldsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
1465    \cyru\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyr\CYRY
1466    \cyrya\CYRYA\cyrat\CYRYAT\cyrhcrs\CYRYHCRS\cyryi\CYRYI\cyryo
1467    \CYRYO\cyryu\CYRYU\cyrz\CYRZ\cyrzdsc\CYRZDSC\cyrrh\CYRRH
1468    \cyrrhdsc\CYRRHDSC}%
1469 \let\update@uclc@with@cyrillic\relax
1470 }
```

Here we process each option:

```
1471 \DeclareOption*{%
1472   \let\encodingdefault\CurrentOption
1473   \edef\reserved@f{%
1474     \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
1475   \reserved@f
1476   \InputIfFileExists\reserved@f
1477   {\PackageError{fontenc}%
1478     {Encoding file '\reserved@f' not found.%
1479     \MessageBreak
1480     You might have misspelt the name of the encoding}%
1481     {Necessary code for this encoding was not
1482     loaded.\MessageBreak
1483     Thus calling the encoding later on will
1484     produce further error messages.}}%
1485   \let\reserved@f\relax

```

In case the current encoding is one of a list of known cyrillic ones we extend the \@uclclist:

```
1486 \expandafter\in@\expandafter{\CurrentOption}%

```

```

1487 {T2A,T2B,T2C,X2,LCY,OT2}%
1488 \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.

```

1489 \expandafter\in@\expandafter\cyr\expandafter
1490 {\@uclclist}%
1491 \ifin@
1492 \else
1493 \update@uclc@with@cyrillic
1494 \fi
1495 \fi
1496 }
1497 \ProcessOptions*

```

We select the new font encoding default (i.e., the last encoding specified in the option list. But this encoding may not work with the current `\f@shape`, e.g., `LY1` is not defined for `cmr` and therefore packages switching to `LY1` usually also change `\rmdefault`. But that only applies at `\begin{document}` so we get a spurious warning if we use what L^AT_EX previously used:

```

1498 %\fontencoding\encodingdefault\selectfont

```

So instead we do this here:

```

1499 \usefont\encodingdefault\familydefault\seriesdefault\shapedefault

```

To save some space we get rid of the macro extending the `\@uclclist` (might have happened already).

```

1500 \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.

```

1501 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
1502 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
1503 \global\let\@ifl@ter@@\@ifl@ter
1504 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
1505 \endpackage

```

File m

ltcounts.dtx

22 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 _E X number (as opposed to <code>\the<counter></code> which expands to the <i>printed</i> representation of <code><counter></code>)
<code>\arabic</code>	<code>\arabic{<counter>}</code> : 1, 2, 3, ...
<code>\roman</code>	<code>\roman{<counter>}</code> : i, ii, iii, ...
<code>\Roman</code>	<code>\Roman{<counter>}</code> : I, II, III, ...
<code>\alph</code>	<code>\alph{<counter>}</code> : a, b, c, ...
<code>\Alph</code>	<code>\Alph{<counter>}</code> : A, B, C, ...
<code>\fnsymbol</code>	<code>\fnsymbol{<counter>}</code> : *, †, ‡, ...
<code>\counterwithin</code>	<code>\counterwithin{<counter>}{<within-counter>}</code> : Resets <code><counter></code> whenever <code><within-counter></code> is stepped. Also redefines <code>\the<counter></code> command to produce <code>\the<within-counter>.\arabic{<counter>}</code> . Star form omits redefining the print representation.
<code>\counterwithout</code>	<code>\counterwithout{<counter>}{<within-counter>}</code> : Removes <code><counter></code> from the reset list of <code><within-counter></code> . Also redefines <code>\the<counter></code> command to produce <code>\arabic{<counter>}</code> . Star form omits redefining the print representation.

1 (*2ekernel)

22.1 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.

`\@removefromreset{<foo>}{<bar>}` : Removes 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 \<latexrelease>\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}}}
45 \</2ekernel>

```

\@removefromreset

```

46 \<latexrelease>\IncludeInRelease{2018-04-01}
47 \<latexrelease>{\@removefromreset}{Add interfaces}%
48 \<*2ekernel | latexrelease>
49 \def\@removefromreset#1#2{%

```

Even through this is internal and the programmer should know what he/she is doing we test here if counter #2 is defined. If not, the execution would run into a tight loop.

```

50 \ifundefined{c@#2}\relax
51 {\begingroup
52 \expandafter\let\csname c@#1\endcsname\@removefromreset
53 \def\@elt##1{%
54 \expandafter\ifx\csname c@##1\endcsname\@removefromreset

```

```

55         \else
56             \noexpand\@elt{##1}%
57         \fi}%
58     \expandafter\xdef\csname cl@#2\endcsname
59     {\csname cl@#2\endcsname}%
60     \endgroup}}

\@ifbothcounters Test if arg #1 and #2 are counters and if so execute #3.
61 \def\@ifbothcounters#1#2#3{%
62     \@ifundefined{c@#1}{\@nocounterr{#1}}%
63     {% else counter is defined
64         \@ifundefined{c@#2}{\@nocounterr{#2}}%
65         {% else both counter and within are defined
66             #3}}}

\counterwithout
67 \def\counterwithout {\@ifstar\counterwithout@s\counterwithout@x}
68 \def\counterwithout@s#1#2{%
69     \@ifbothcounters{#1}{#2}{\@removefromreset{#1}{#2}}}
70 \def\counterwithout@x#1#2{%
71     \@ifbothcounters{#1}{#2}%
72     {\@removefromreset{#1}{#2}%
73         \expandafter
74         \gdef\csname the#1\expandafter\endcsname\expandafter
75             {\expandafter
76                 \@arabic\csname c@#1\endcsname}}}}

\counterwithin
77 \def\counterwithin{\@ifstar\counterwithin@s\counterwithin@x}
78 \def\counterwithin@s#1#2{%
79     \@ifbothcounters{#1}{#2}{\@addtoreset{#1}{#2}}}
80 \def\counterwithin@x#1#2{%
81     \@ifbothcounters{#1}{#2}%
82     {\@addtoreset{#1}{#2}%
83         \expandafter
84         \gdef\csname the#1\expandafter\endcsname\expandafter
85             {\csname the#2\expandafter\endcsname\expandafter
86                 .\expandafter
87                 \@arabic\csname c@#1\endcsname}}}}

88 </2ekernel | latexrelease>
89 <latexrelease>\EndIncludeInRelease
90 <latexrelease>\IncludeInRelease{0000-00-00}
91 <latexrelease>         {\@removefromreset}{Add interfaces}%
92 <latexrelease>\let \@removefromreset \undefined
93 <latexrelease>\let \@ifbothcounters \undefined
94 <latexrelease>\let \counterwithout \undefined
95 <latexrelease>\let \counterwithout@s \undefined
96 <latexrelease>\let \counterwithout@x \undefined
97 <latexrelease>\let \counterwithin \undefined
98 <latexrelease>\let \counterwithin@s \undefined
99 <latexrelease>\let \counterwithin@x \undefined

```

```

100 (latexrelease)\EndIncludeInRelease
101 (*2ekernel)

    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.
102 \def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}

\roman Representation of counter as lower-case Roman numerals.
103 \def\roman#1{\expandafter\@roman\csname c@#1\endcsname}

\Roman Representation of counter as upper-case Roman numerals.
104 \def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}

\alph Representation of counter as a lower-case letter: 1 = a, 2 = b, etc.
105 \def\alph#1{\expandafter\@alph\csname c@#1\endcsname}

\Alph Representation of counter as an upper-case letter: 1 = A, 2 = B, etc.
106 \def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}

\fnsymbol Representation of COUNTER as a footnote symbol: 1 = *, 2 = †, etc.
107 \def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}

\@arabic \@arabic\F00counter Representation of \F00counter as arabic numerals.
108 \def\@arabic#1{\number #1} %% changed 29 Apr 86

\@roman \@roman\F00counter Representation of \F00counter as lower-case Roman nu-
merals.
109 \def\@roman#1{\romannumeral #1}

\@Roman \@Roman\F00counter Representation of \F00counter as upper-case Roman nu-
merals.
110 \def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}

\@slowromancap Fully expandable macro to change a roman number to uppercase.
111 \def\@slowromancap#1{\ifx @#1% then terminate
112     \else
113         \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
114             c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi
115             \expandafter\@slowromancap
116         \fi
117 }

\@alph \@alph\F00counter Representation of \F00counter as a lower-case letter: 1 =
a, 2 = b, etc.
118 \def\@alph#1{%
119     \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
120     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
121     y\or z\else\@ctrerr\fi}

```


`\@Alph` `\@Alph\FOOcounter` Representation of `\FOOcounter` as an upper-case letter: 1 = A, 2 = B, etc.

```
122 \def\@Alph#1{%
123   \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
124   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
125   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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$: Determining 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, which ruins any kerning between the preceding characters and whatever awaits typesetting. If you use $\mathrm{eT}_{\mathrm{E}}\mathrm{X}$ as engine for $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ (as recommended) this unfortunate side effect is not present.

```
126 </2ekernel>
127 <latexrelease>\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TextOrMath}%
128 <*2ekernel | latexrelease>
129 \def\@fnsymbol#1{%
130   \ifcase#1\or \TextOrMath\textasteriskcentered *\or
131   \TextOrMath \textdagger \dagger\or
132   \TextOrMath \textdaggerdbl \ddagger \or
133   \TextOrMath \textsection \mathsection\or
134   \TextOrMath \textparagraph \mathparagraph\or
135   \TextOrMath \textbardbl ||\or
136   \TextOrMath {\textasteriskcentered\textasteriskcentered}{**}\or
137   \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
138   \TextOrMath {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
139   \@ctrerr \fi
140 }%
141 </2ekernel | latexrelease>
142 <latexrelease>\EndIncludeInRelease
143 <latexrelease>\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TextOrMath}%
144 <latexrelease>\def\@fnsymbol#1{\ensuremath{%
145 <latexrelease>   \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
146 <latexrelease>   \mathparagraph\or ||\or **\or \dagger\dagger
147 <latexrelease>   \or \ddagger\ddagger \else\@ctrerr\fi}}%
148 <latexrelease>\EndIncludeInRelease
149 <*2ekernel>
```

`\TextOrMath` When using regular $\mathrm{T}_{\mathrm{E}}\mathrm{X}$, we make this command robust so that it always selects the correct branch in an `\ifmode` 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 e \TeX 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 e \TeX but making sure not to permanently turn `\protected` into `\relax`.

```
150 </2ekernel>
151 <latexrelease>\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
152 <*2ekernel | latexrelease>
153 \begingroup\expandafter\expandafter\expandafter\endgroup
154 \expandafter\ifx\csname protected\endcsname\relax
```

In case of ordinary \TeX 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.

```
155 \DeclareRobustCommand\TextOrMath{%
156   \ifmmode   \expandafter\@secondoftwo
157   \else      \expandafter\@firstoftwo   \fi}
158 \protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
159 \else
```

For e \TeX the situation is similar. The robust macro is a hidden one so that we again avoid problems of gobbling spaces in the input.

```
160 \protected\expandafter\def\csname TextOrMath\space\endcsname{%
161   \ifmmode   \expandafter\@secondoftwo
162   \else      \expandafter\@firstoftwo   \fi}
163 \edef\TextOrMath#1#2{%
164   \expandafter\noexpand\csname TextOrMath\space\endcsname
165   {#1}{#2}}
166 \fi
167 </2ekernel | latexrelease>
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
170 <latexrelease>\let\TextOrMath\@undefined
171 <latexrelease>\EndIncludeInRelease
172 <*2ekernel>

173 </2ekernel>
```

File n ltnlength.dtx

23 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 \<latexrelease>\IncludeInRelease{2015/01/01}%
6 \<latexrelease> \setlength}{Using \setlength with \dimen0}%
7 (*2ekernel | latexrelease)

8 \def\setlength#1#2{#1 #2\relax}
9 (/2ekernel | latexrelease)
10 \<latexrelease>\EndIncludeInRelease
11 \<latexrelease>\IncludeInRelease{0000/00/00}%
12 \<latexrelease> \setlength}{Using \setlength with \dimen0}%
13 \<latexrelease>\def\setlength#1#2{#1#2\relax}
14 \<latexrelease>\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 17 \def\@settodim#1#2#3{\setbox\@tempboxa\hbox{#{#3}}#2#1\@tempboxa
\settowidth Clear the memory afterwards (which might be a lot).
\@settodim 18 \setbox\@tempboxa\box\voidb@x}
19 \DeclareRobustCommand\settoheight{\@settodim\ht}
20 \DeclareRobustCommand\settodepth {\@settodim\dp}
21 \DeclareRobustCommand\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

lftssbas.dtx

This file contains the main implementation of the ‘low level’ font selection commands. See other parts of the L^AT_EX distribution, or *The L^AT_EX Companion* for higher level documentation of the L^AT_EX ‘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.

24 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.

```
1 (*2kernel)
2 \def\@nomath#1{\relax\ifmmode
3   \@font@warning{Command \@nomath#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_EX from scanning too far in certain situations.

```
4 \gdef\no@alphabet@error#1{\relax \ifmmode
5   \@latex@error{Math\space alphabet\space identifier\space
6     \noexpand#1is\space undefined\space in\space math\space
7     version\space ‘\math@version’}%
8   {Your\space requested\space math\space alphabet\space
9     is\space undefined\space in\space the\space current\space
10    math\space version.^^JCheck\space the\space spelling\space
11    or\space use\space the\space \noexpand\SetMathAlphabet\space
12    command.}
13   \fi}
```

`\new@mathgroup` We also give a new name to `\newfam` and `\fam` to avoid verbal confusion (see the introduction).²

`\mathgroup`

```
14 %\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@@n}
15 \let\mathgroup\fam
16 %\let\newfam\new@mathgroup
17 \@onlypreamble\new@mathgroup
```

²For 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.

25 Macros for setting up the tables

`\DeclareFontShape` The macro `\DeclareFontShape` takes 6 arguments:

```
18 \def\DeclareFontShape{\begingroup
```

First we restore the catcodes of all characters used in the syntax.

```
19 \nfss@catcodes
```

We use `\expandafter \endgroup` to restore catcode in case something goes wrong with the argument parsing (suggested by Tim Van Zandt)

`\DeclareFontShape`

```
20 \expandafter\endgroup
21 \DeclareFontShape@}
22 \def\DeclareFontShape@#1#2#3#4#5#6{%
23 \expandafter\ifx\csname #1+#2\endcsname\relax
24 \latex@error{Font family ‘#1+#2’ unknown}\@eha
25 \else
26 \expandafter
27 \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
28 \csname #5\endcsname}%
29 \def\reserved@a{#6}%
30 \global
31 \expandafter\let\csname#5\endcsname\expandafter\endcsname
32 \ifx\reserved@a@empty
33 \@empty
34 \else
35 \reserved@a
36 \fi
37 \fi
38 }
```

`\DeclareFixedFont` Define a direct font switch that avoids all overhead.

```
39 \def\DeclareFixedFont#1#2#3#4#5#6{%
40 \begingroup
41 \math@fontsfalse
42 \every@math@size{}%
43 \fontsize{#6}\z@
44 \usefont{#2}{#3}{#4}{#5}%
45 \global\expandafter\let\expandafter#1\the\font
46 \endgroup
47 }
```

`\do@subst@correction`

```
48 \def\do@subst@correction{%
49 \xdef\subst@correction{%
50 \font@name
51 \global\expandafter\font
52 \csname \curr@fontshape/\f@size\endcsname
53 \noexpand\fontname\font
54 \relax}%

```

Calling `\subst@correction` after the current group means calling it after we have loaded the substitution font which is done inside a group.

```
55 \aftergroup\subst@correction
56 }
```

`\DeclareFontFamily`

```
57 \def\DeclareFontFamily#1#2#3{%
```

If we want fast checking for the encoding scheme we can just check for `\T@.` being defined.

```
58 % \@tempswafalse
59 % \def\reserved@b{#1}%
60 % \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
61 %     \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
62 % \cdp@list
63 % \if@tempswa
64 % \ifundefined{T@#1}%
65 %     {%
66 %         \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
67 %     }%
68 % }
```

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`.

```
69     \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`.

```
70     \global
71     \expandafter\let\csname #1+#2\expandafter\endcsname
72         \ifx \reserved@a\@empty
73             \@empty
74         \else \reserved@a
75         \fi
76     }%
77 }
```

`\cdp@list` We initialize the code page list to be empty.

```
78 \let\cdp@list\@empty
79 \@onlypreamble\cdp@list
```

`\cdp@elt`

```
80 \let\cdp@elt\relax
81 \@onlypreamble\cdp@elt
```

`\DeclareFontEncoding`

```
82 \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.

```
83     \begingroup
84     \nfss@catcodes
85     \expandafter\endgroup
86     \DeclareFontEncoding@}
87 \@onlypreamble\DeclareFontEncoding
```

```

88 \def\DeclareFontEncoding#1#2#3{%
89   \expandafter
90   \ifx\csname T@#1\endcsname\relax
91     \def\cdp@elt{\noexpand\cdp@elt}%
92     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
93                   {\default@family}{\default@series}%
94                   {\default@shape}}%

```

To support encoding dependent commands (like accents) we initialise the command `\<encoding>-cmd` to be `\@changed@cmd`. (See `ltoutenc.dtx` for details.)

```

95   \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
96   \else
97     \font@info{Redeclaring font encoding #1}%
98   \fi
99   \global\@namedef{T@#1}{#2}%
100  \global\@namedef{M@#1}{\default@M#3}%

```

Keep a record of the last encoding being declared:

```

101  \xdef\LastDeclaredEncoding{#1}%
102  }
103 \@onlypreamble\DeclareFontEncoding@

```

`\LastDeclaredEncoding` The last encoding being declared by `\DeclareFontEncoding`.

```

104 \def\LastDeclaredEncoding{}

```

`\DeclareFontSubstitution`

```

105 \def\DeclareFontSubstitution#1#2#3#4{%
106   \expandafter
107   \ifx\csname T@#1\endcsname\relax
108     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
109   \else
110     \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`.

```

111     \edef\reserved@a{#1}%
112     \toks@{}%
113     \def\cdp@elt##1##2##3##4{%
114       \def\reserved@b{##1}%
115       \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.

```

116       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
117     \else

```

If `\reserved@a` and `\reserved@b` differ then we simply copy from the old list to the new.

```

118       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
119     \fi}%
120     \cdp@list

```

```

121     \xdef\cdp@list{\the\toks@}%
122   \endgroup
123   \global
124   \@namedef{D@#1}{%
125     \def\default@family{#2}%
126     \def\default@series{#3}%
127     \def\default@shape{#4}%
128   }%
129 \fi
130 }
131 \@onlypreamble\DeclareFontSubstitution

```

\DeclareFontEncodingDefaults

```

132 \def\DeclareFontEncodingDefaults#1#2{%
133   \ifx\relax#1\else
134     \ifx\default@T\@empty\else
135       \@font@info{Overwriting encoding scheme text defaults}%
136     \fi
137     \gdef\default@T{#1}%
138   \fi
139   \ifx\relax#2\else
140     \ifx\default@M\@empty\else
141       \@font@info{Overwriting encoding scheme math defaults}%
142     \fi
143     \gdef\default@M{#2}%
144   \fi
145 }
146 \@onlypreamble\DeclareFontEncodingDefaults

```

\default@T

\default@M

```

147 \let\default@T\@empty
148 \let\default@M\@empty

```

\DeclarePreloadSizes

```

149 \def\DeclarePreloadSizes#1#2#3#4#5{%
150   \@ifundefined{T@#1}%
151   {\@latex@error{Encoding scheme ‘#1’ unknown}\@eha}%
152   {%

```

Don't know at the moment what this group here does!

```

153   \begingroup

```

We define a macro `\reserved@f`³ that grabs the next *size* and loads the corresponding font. This is done by delimiting `\reserved@f`'s only argument by the token `,` (comma).

```

154   \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_EXbook*: 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.)

```

155     \if>##1>%

```

³We cannot use `\@tempa` since it is needed in `\pickup@font`.


```

156         \let\reserved@f\relax
157     \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.

```

158         \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
159         \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.

```

160         \global\expandafter\let\font@name\relax
161     \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.

```

162     \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.

```

163     \reserved@f#5,,%
164     \endgroup
165 }%
166 }
167 \@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.

```

168 \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.

```

169 \def\DeclareMathSizes{%
170     \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
171     {\@DeclareMathSizes{}}%
172 \@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}`.

```

173 \ifx\@onlypreamble\@onlypreamble\relax\relax
174 \ifx\@onlypreamble\@onlypreamble\relax\relax
175 \ifx\@onlypreamble\@onlypreamble\relax\relax
176 \ifx\@onlypreamble\@onlypreamble\relax\relax
177 \def\@DeclareMathSizes #1#2#3#4#5{%
178     \@defaultunits\dimen@ #2pt\relax\@nnil
179     \if $3$%
180         \expandafter\let\csname S@strip@pt\dimen@\endcsname\math@fontsfalse
181     \else

```

```

182 \@defaultunits\dimen@ii #3pt\relax\@nnil
183 \@defaultunits\@tempdima #4pt\relax\@nnil
184 \@defaultunits\@tempdimb #5pt\relax\@nnil
185 \toks@{#1}%
186 \expandafter\edef\csname S@\strip@pt\dimen@\endcsname{%
187 \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
188 \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
189 \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
190 \the\toks@
191 }%
192 \fi
193 }%
194 \ifx\kernel\@latexrelease
195 \EndIncludeInRelease
196 \ifx\@latexrelease\@includeinrelease{0000/00/00}\@DeclareMathSizes}%
197 \ifx\@latexrelease\@includeinrelease{Arbitrary units in \DeclareMathSizes}%
198 \ifx\@latexrelease\@includeinrelease{\def\@DeclareMathSizes#1#2#3#4#5{%
199 \ifx\@defaultunits\dimen@#2pt\relax\@nnil
200 \if$#3$%
201 \expandafter\let
202 \csname S@\strip@pt\dimen@\endcsname
203 \math@fontsfalse
204 \else
205 \expandafter\gdef
206 \csname S@\strip@pt\dimen@\endcsname
207 {\gdef\tf@size{#3}\gdef\sf@size{#4}%
208 \gdef\ssf@size{#5}%
209 #1%
210 }%
211 \fi}%
212 \ifx\@latexrelease\@includeinrelease
213 \ifx\kernel\@latexrelease
214 \onlypreamble\@DeclareMathSizes

```

26 Selecting a new font

26.1 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.

```

215 \DeclareRobustCommand\fontencoding[1]{%
216 \expandafter\ifx\csname T@#1\endcsname\relax
217 \latexerror{Encoding scheme ‘#1’ unknown}\@eha
218 \else
219 \edef\f@encoding{#1}%
220 \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`.

```

221      \let\enc@update\relax
222      \else

```

If current and new encoding differ we define the macro `\enc@update` to contain all updates necessary at `\selectfont` time.

```

223      \let\enc@update\@@enc@update
224      \fi
225      \fi
226  }

```

`\@@enc@update`

```

227 \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`.

```

228      \expandafter
229      \let
230      \csname\cf@encoding -cmd\endcsname
231      \@changed@cmd

```

Then we turn the one for the new encoding to `\@current@cmd` (see `ltoutenc.dtx` for further explanations).

```

232      \expandafter
233      \let
234      \csname\f@encoding-cmd\endcsname
235      \@current@cmd

```

We execute the default settings `\default@T`, followed by the one for the new encoding.

```

236      \default@T
237      \csname T@\f@encoding\endcsname

```

Finally we change the default substitution values, disable `\enc@update` and make `\f@encoding` officially the current encoding.

```

238      \csname D@\f@encoding\endcsname
239      \let\enc@update\relax
240      \let\cf@encoding\f@encoding
241  }

```

`\enc@update` The default action in `\selectfont` is to do nothing.

```

242 \let\enc@update\relax

```

`\fontfamily`

```

\fontfamily 243 \DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}

```

`\fontseries` There are now defined later (and differently).

```

\fontseries 244 %\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}

```

```

\fontshape 245 %\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}

```

`\f@shape`

`\usefont` 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.

`\fontencoding` needs to do some setup work so we call that, but instead of calling `\fontfamily`, `\fontseries` and `\fontshape` it earlier versions of this

code did, we now set `\f@family`, etc. directly. If we would call `\fontseries` or `\fontshape` as it was done in the past, they would now interact with the existing series and shape which is not desired if we intend to use an explicit font shape!

```
246 \DeclareRobustCommand\usefont[4]{\fontencoding{#1}%
247   \edef\f@family{#2}%
248   \edef\f@series{#3}%
249   \edef\f@shape{#4}\selectfont
250   \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.

```
251 \DeclareRobustCommand\linespread[1]
252   {\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.

```
253 \DeclareRobustCommand\fontsize[2]
254   {\set@fontsize\baselinestretch{#1}{#2}}
```

`\f@linespread` This macro holds the current internal value for `\baselinestretch`.

```
255 \let\f@family\@empty
256 \let\f@series\@empty
257 \let\f@shape\@empty
258 \let\f@size\@empty
259 \let\f@baselineskip\@empty
260 \let\f@linespread\@empty
```

`\cf@encoding`

```
261 \let\f@encoding\@empty
262 \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.

```
263 \def\@defaultunits{\afterassignment\remove@to@nnil}
```

`\strip@pt` This macro strips the characters `pt` produced by using `\the` on a dimen register.

```
\rem@pt
264 \begingroup
265   \catcode‘P=12
266   \catcode‘T=12
267   \lowercase{
```

```

268 \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
269 \expandafter\endgroup\x
270 \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.

```

271 \DeclareRobustCommand\mathversion[1]
272     {\@nomath\mathversion
273     \expandafter\ifx\csname mv@#1\endcsname\relax
274     \@latex@error{Math version ‘#1’ is not defined}\@eha\else
275     \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@currsz` to an invalid value since this will trigger the setup when the formula starts.

```

276 \gdef\glb@currsz{}%

```

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`.

```

277 \aftergroup\glb@settings
278 \fi}

```

If $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ the use of `$` (as the primitive $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ function must be hidden.

Anyway, since we don't have this and won't have it in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ 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 279 \let\frozen@everymath\everymath
280 \let\frozen@everydisplay\everydisplay

```

```

\everymath Now we provide now user hooks that will be called in the frozen internals.
\everydisplay 281 \newtoks\everymath
282 \newtoks\everydisplay

```

```

\frozen@everymath Now we define the behaviour of the frozen hooks: first check the math setup then
call the user hook.
283 \frozen@everymath = {\check@mathfonts
284 \the\everymath}

```

`\frozen@everydisplay` Ditto for the display hook.

```

285 \frozen@everydisplay = {\check@mathfonts
286                          \the\everydisplay}
```

`\curr@math@size` This holds locally the current math size.

```

287 \let\curr@math@size\@empty
```

26.2 Macros for loading fonts

`\pickup@font` The macro `\pickup@font` which is used in `\selectfont` is very simple: if the font name is undefined (i.e. not known yet) it calls `\define@newfont` to load it.

```

288 \def\pickup@font{%
289     \expandafter \ifx \font@name \relax
290         \define@newfont
291     \fi}
```

`\split@name` `\pickup@font` assumes that `\font@name` is set but it is sometimes called when `\f@family`, `\f@series`, `\f@shape`, or `\f@size` may have the wrong settings (see, e.g., the definition of `\getanddefine@fonts`). Therefore we need a macro to extract font *family*, *series*, *shape*, and *size* from the font name. To this end we define `\split@name` which takes the font name as a list of characters of `\catcode` 12 (without the backslash at the beginning) delimited by the special control sequence `\@nil`. This is not very complicated: we first ensure that `/` has the right `\catcode`

```

292 {\catcode'\/=12
```

and define `\split@name` so that it will define our private `\f@encoding`, `\f@family`, `\f@series`, `\f@shape`, and `\f@size` macros.

```

293 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}%
294                                     \def\f@family{#2}%
295                                     \def\f@series{#3}%
296                                     \def\f@shape{#4}%
297                                     \def\f@size{#5}}}
```

`\curr@fontshape` Abbreviation which may get removed again for speed.

```

298 \def\curr@fontshape{\f@encoding/\f@family/\f@series/\f@shape}
```

`\define@newfont` Now we can tackle the problem of defining a new font.

```

299 \def\define@newfont{%
```

We have already mentioned that the token list that `\split@name` will get as argument must not start with a backslash. To reach this goal we will set the `\escapechar` to `-1` so that the `\string` primitive will not generate an escape character. To keep this change local we open a group. We use `\begingroup` for this purpose since `\define@newfont` might be called in math mode, and an empty `\bgroup... \egroup` would add an empty Ord atom to the math list and thus affect the spacing.

Also locally redefine `\typeout` so that ‘No file ...fd’ Warnings become Font Info message just sent to the log file.

```

300 \begingroup
301     \let\typeout\@font@info
302     \escapechar\m@ne
```

Then we extract *encoding scheme*, *family*, *series*, *shape*, and *size* from the font name. Note the four `\expandafter`'s so that `\font@name` is expanded first, then `\string`, and finally `\split@name`.

```
303 \expandafter\expandafter\expandafter
304 \split@name\expandafter\string\font@name\@nil
```

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.

```
305 % \expandafter\ifx
306 % \csname\curr@fontshape\endcsname \relax
307 \try@load@fontshape % try always
308 % \fi
309 \expandafter\ifx
310 \csname\curr@fontshape\endcsname \relax
311 \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).

```
312 % \csname\curr@fontshape\endcsname
313 \extract@font\fi
```

We are nearly finished and must only restore the `\escapechar` by closing the group.

```
314 \endgroup}
```

```
315 \def\try@load@fontshape{%
316 \expandafter
317 \ifx\csname \f@encoding+\f@family\endcsname\relax
318 \font@info{Trying to load font information for
319 \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.

```
320 \global\expandafter\let
321 \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).

```
322 \nfss@catcodes
323 \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.

```
324 \edef\reserved@a{%
325 \lowercase{%
326 \noexpand\inputIfFileExists{\f@encoding\f@family.fd}}}%
327 \reserved@a\relax
328 {\@input@\f@encoding\f@family.fd}%
329 \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.

```
330 \def\nfss@catcodes{%
```

We start by making `@` a letter and ignoring all blanks and newlines.

```
331 \makeatletter
332 \catcode'\ 9%
333 \catcode'\^^I9%
334 \catcode'\^^M9%
```

Then we set up `\`, `{`, `}`, `#` and `%` in case an `.fd` file is loaded during a verbatim environment.

```
335 \catcode'\z@
336 \catcode'\{ \@ne
337 \catcode'\}\ @tw@
338 \catcode'\#6%
339 \catcode'\^7%
340 \catcode'\%14%
```

Then we make sure that the important syntax parts have the right `\catcode`.

```
341 \@makeother\<%
342 \@makeother\>%
343 \@makeother\*%
344 \@makeother\.%
345 \@makeother\-%
346 \@makeother\/%
347 \@makeother\[%
348 \@makeother\]%
349 \@makeother\'%
350 \@makeother\'%
351 \@makeother\"%
352 }
```

`\LoadFontDefinitionFile` Load and `.fd` files for some encoding and family (if it exists).

```
353 </2ekernel>
354 <*2ekernel | latexrelease>
355 <latexrelease>\IncludeInRelease{2020/02/02}%
356 <latexrelease>{\LoadFontDefinitionFile}{Loading .fd files}%
357 \def\LoadFontDefinitionFile#1#2{%
358 \begingroup
359 \edef\f@encoding{#1}%
360 \edef\f@family{#2}%
361 \try@load@fontshape
362 \endgroup
363 }
364 </2ekernel | latexrelease>
```



```

365 \latexrelease\EndIncludeInRelease
366 \latexrelease\IncludeInRelease{0000/00/00}%
367 \latexrelease\LoadFontDefinitionFile\{Loading .fd files}%
368 \latexrelease\let\LoadFontDefinitionFile\@undefined
369 \latexrelease\EndIncludeInRelease
370 \latexrelease\EndIncludeInRelease
371 \*2ekernel)

```

DeclareFontFamilySubstitution The idea for this macro is stolen from the `substitutefont` package by Günter Milde, with some modifications and a new name.

Its purpose is to provide characters in a special encoding tht are not available in the current font family to be taken from a different family that is visually compatible (or not if you choose badly). For example, you can match the GFS Didot Greek characters with T_EX Gyre Pagella (Palatino) by specifying

```
\DeclareFontFamilySubstitution{LGR}{qpl}{udidot}
```

This way if you ask for the LGR encoding in for the qpl family you get the characters from the udidot family substituted.

We need to ensure that the macro is defined with `\nfss@catcodes` in force (not quite sure why at the moment to be honest).

```

372 \endkernel)
373 \*2ekernel | latexrelease)
374 \latexrelease\IncludeInRelease{2020/02/02}%
375 \latexrelease\DeclareFontFamilySubstitution\{Provide family substitution}%
376 \begingroup
377 \nfss@catcodes
378 \gdef\DeclareFontFamilySubstitution#1#2#3{%

```

We only provide a set of silent substitutions. The package also (re)declared the family, but this is incorrect in my eyes and it is better to handle that differently.

Of course the families may still need loading at this point and so we arrange for this. Otherwise we might run into trouble because the necessary `\DeclareFontFamily` has not been seen.

```

379 \LoadFontDefinitionFile{#1}{#2}%
380 \LoadFontDefinitionFile{#1}{#3}%
381 \DeclareFontShape{#1}{#2}{m}{it}{<->ssub * #3/m/it}{}%
382 \DeclareFontShape{#1}{#2}{m}{n}{<->ssub * #3/m/n}{}%
383 \DeclareFontShape{#1}{#2}{m}{sc}{<->ssub * #3/m/sc}{}%
384 \DeclareFontShape{#1}{#2}{m}{sl}{<->ssub * #3/m/sl}{}%

```

These days a few more shapes might be around, so we declare those too. If they don't exist then after the first substitution normal fallbacks will happen.

```

385 \DeclareFontShape{#1}{#2}{m}{sw}{<->ssub * #3/m/sw}{}%
386 \DeclareFontShape{#1}{#2}{m}{scit}{<->ssub * #3/m/scit}{}%
387 \DeclareFontShape{#1}{#2}{m}{scsl}{<->ssub * #3/m/scsl}{}%

```

Same game with b and bx, for other weights you are on your own:

```

388 \DeclareFontShape{#1}{#2}{b}{it}{<->ssub * #3/b/it}{}%
389 \DeclareFontShape{#1}{#2}{b}{n}{<->ssub * #3/b/n}{}%
390 \DeclareFontShape{#1}{#2}{b}{scit}{<->ssub * #3/b/scit}{}%
391 \DeclareFontShape{#1}{#2}{b}{scsl}{<->ssub * #3/b/scsl}{}%
392 \DeclareFontShape{#1}{#2}{b}{sc}{<->ssub * #3/b/sc}{}%
393 \DeclareFontShape{#1}{#2}{b}{sl}{<->ssub * #3/b/sl}{}%

```

```

394 \DeclareFontShape{#1}{#2}{b}{sw}{<->ssub * #3/b/sw}{}%
395 \DeclareFontShape{#1}{#2}{bx}{it}{<->ssub * #3/bx/it}{}%
396 \DeclareFontShape{#1}{#2}{bx}{n}{<->ssub * #3/bx/n}{}%
397 \DeclareFontShape{#1}{#2}{bx}{scit}{<->ssub * #3/bx/scit}{}%
398 \DeclareFontShape{#1}{#2}{bx}{scsl}{<->ssub * #3/bx/scsl}{}%
399 \DeclareFontShape{#1}{#2}{bx}{sc}{<->ssub * #3/bx/sc}{}%
400 \DeclareFontShape{#1}{#2}{bx}{sl}{<->ssub * #3/bx/sl}{}%
401 \DeclareFontShape{#1}{#2}{bx}{sw}{<->ssub * #3/bx/sw}{}%
402 }
403 \endgroup
404 /<2ekernel | latexrelease>
405 <latexrelease>\EndIncludeInRelease
406 <latexrelease>\IncludeInRelease{0000/00/00}%
407 <latexrelease> \{\DeclareFontFamilySubstitution\}{Provide family substitution}%
408 <latexrelease>
409 <latexrelease>\let\DeclareFontFamilySubstitution\@undefined
410 <latexrelease>\EndIncludeInRelease
411 <*2ekernel>

```

`\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!

```

412 /<2ekernel>
413 <*2ekernel | latexrelease>
414 <latexrelease>\IncludeInRelease{2019/10/01}%
415 <latexrelease> \{\DeclareErrorFont\}{No side effects please}%
416 \def\DeclareErrorFont#1#2#3#4#5{%
417     \xdef\error@fontshape{%
418         \noexpand\expandafter\noexpand\split@name\noexpand\string
419         \expandafter\noexpand\cname#1/#2/#3/#4/#5\endcsname
420         \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—and now it did.

```

421 %     \gdef\f@encoding{#1}%
422     \gdef\default@family{#2}%
423     \gdef\default@series{#3}%
424     \gdef\default@shape{#4}%
425 }
426 /<2ekernel | latexrelease>
427 <latexrelease>\EndIncludeInRelease
428 <latexrelease>\IncludeInRelease{0000/00/00}%
429 <latexrelease> \{\DeclareErrorFont\}{No side effects please}%
430 <latexrelease>
431 <latexrelease>\def\DeclareErrorFont#1#2#3#4#5{%
432 <latexrelease> \xdef\error@fontshape{%
433 <latexrelease> \noexpand\expandafter\noexpand\split@name\noexpand\string
434 <latexrelease> \expandafter\noexpand\cname#1/#2/#3/#4/#5\endcsname
435 <latexrelease> \noexpand\@nil}%
436 <latexrelease> \gdef\default@family{#2}%

```

```

437 \latexrelease) \gdef\default@series{#3}%
438 \latexrelease) \gdef\default@shape{#4}%
439 \latexrelease) \global\let\f@family\default@family
440 \latexrelease) \global\let\f@series\default@series
441 \latexrelease) \global\let\f@shape\default@shape
442 \latexrelease) \gdef\f@size{#5}%
443 \latexrelease) \gdef\f@baselineskip{#5pt}%
444 \latexrelease)}
445 \latexrelease)\EndIncludeInRelease
446 (*2ekernel)

447 \@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_EX will go into an infinite loop. But if the defaults are set incorrectly one deserves nothing else!

```

448 \endkernel)
449 \latexrelease)\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
450 \latexrelease) {Font substitution in preamble}%
451 (*2ekernel | latexrelease)
452 \def\wrong@fontshape{%
453   \csname D@f@encoding\endcsname % install defaults if in math
454   \edef\reserved@a{\csname\curr@fontshape\endcsname}%
455   \ifx\last@fontshape\reserved@a
456     \errmessage{Corrupted NFSS tables}%
457     \error@fontshape
458   \else

```

Then we warn the user about the mess and set the shape to its default.

```

459   \let\f@shape\default@shape

```

If the combination is not known, try the default *series*.

```

460   \expandafter\ifx\csname\curr@fontshape\endcsname\relax
461     \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!

```

462   \expandafter
463   \ifx\csname\curr@fontshape\endcsname\relax
464     \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}`.

```

465     \begingroup
466     \try@load@fontshape
467   \endgroup
468   \fi \fi
469 \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 TeX 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.

```
470 \font@warning{Font shape ‘\expandafter\string\reserved@a’
471 \expandafter\@gobble\string\undefined\MessageBreak
472 using ‘\curr@fontshape’ instead\@wrong@font@char}%
473 \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`.

```
474 \gdef\@defaultsubs{%
475 \font@warning{Some font shapes were not available, defaults
476 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.

```
477 \global\expandafter\expandafter\expandafter\let
478 \expandafter\reserved@a
479 \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.

```
480 \xdef\font@name{%
481 \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.

```
482 \pickup@font}
483 /2ekernel | latexrelease)
484 (latexrelease)\EndIncludeInRelease
485 (latexrelease)\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
486 (latexrelease) {Font substitution in preamble}%
487 (latexrelease)\def\wrong@fontshape{%
488 (latexrelease) \csname D@\f@encoding\endcsname
489 (latexrelease) \edef\reserved@a{\csname\curr@fontshape\endcsname}%
490 (latexrelease) \ifx\last@fontshape\reserved@a
491 (latexrelease) \errmessage{Corrupted NFSS tables}%
492 (latexrelease) \error@fontshape
493 (latexrelease) \else
494 (latexrelease) \let\f@shape\default@shape
495 (latexrelease) \expandafter\ifx\csname\curr@fontshape\endcsname\relax
```

```

496 \latexrelease> \let\f@series\default@series
497 \latexrelease> \expandafter
498 \latexrelease> \ifx\csname\curr@fontshape\endcsname\relax
499 \latexrelease> \let\f@family\default@family
500 \latexrelease> \fi \fi
501 \latexrelease> \fi
502 \latexrelease> \@font@warning{Font shape
503 \latexrelease> ' \expandafter\string\reserved@a'
504 \latexrelease> \expandafter\@gobble\string\@undefined
505 \latexrelease> \MessageBreak
506 \latexrelease> using '\curr@fontshape' instead\@wrong@font@char}%
507 \latexrelease> \global\let\last@fontshape\reserved@a
508 \latexrelease> \gdef\@defaultsubs{%
509 \latexrelease> \@font@warning{Some font shapes were not available,
510 \latexrelease> defaults substituted.\@gobbletwo}}%
511 \latexrelease> \global\expandafter\expandafter\expandafter\let
512 \latexrelease> \expandafter\reserved@a
513 \latexrelease> \csname\curr@fontshape\endcsname
514 \latexrelease> \xdef\font@name{%
515 \latexrelease> \csname\curr@fontshape/\f@size\endcsname}%
516 \latexrelease> \pickup@font}
517 \latexrelease> \EndIncludeInRelease
518 (*2ekernel)

```

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

```
519 \let\@wrong@font@char\@empty
```

`\@@defaultsubs` See above.

```
\@defaultsubs 520 \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 *TeXbook*, p. 382).

```
521 \def\strip@prefix#1>{\}
```

27 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

```
522 \let\install@mathalphabet\gdef
```

`\math@fonts`

```
523 \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.

```
524 %\def\select@group#1#2#3{\relax\ifmmode
```

We do these things locally using `\begingroup` instead of `\bgroup` to avoid the appearance of an empty Ord atom on the math list.

```
525 % \begingroup
```

We set the math fonts for the *family* in question by calling `\getanddefine@fonts` in the correct environment.

```
526 %      \escapechar\m@ne
527 %      \getanddefine@fonts{\csname c@mv@math@version\endcsname}#3%
```

We globally select the math fonts...

```
528 %      \globaldefs\@ne \math@fonts
```

... and close the group to restore `\globaldefs` and `\escapechar`.

```
529 %      \endgroup
```

As long as no *size* or *version* change occurs the *math alphabet identifier* should simply switch to the installed *math group* instead of calling `\select@group` unnecessarily. So we globally redefine the first argument (the new *math alphabet identifier*) 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 *math alphabet identifier* and then calling the new definition to switch to the wanted *math group*. Now we define the *math alphabet identifier* as a call to the `\use@mathgroup` command.

```
530 %      \xdef#1{\noexpand\use@mathgroup\noexpand#2%
531 %              {\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.

```
532 %      \expandafter\extract@alph@from@version
533 %              \csname mv@math@version\expandafter\endcsname
534 %      \expandafter{\number\csname c@mv@math@version\endcsname}%
535 %      #1%
536 %      \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.

```
537 %\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.

538 `\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`.

539 `\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`.

540 `\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.

541 `\def\reserved@c####1{\gdef#1{##1####1##3}}%`

Then we execute our auxiliary macro.

542 `\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.

543 `\def\reserved@a\select@group#3##1##2\@nil{%`

This macro can now directly rebuild the math version definition by calling `\reserved@c`:

544 `\reserved@c{%`
545 `\getanddefine@fonts{#2}##2%`
546 `\install@mathalphabet#3{%`
547 `\relax\ifmmode \else \non@alpherr#3\fi`
548 `\use@mathgroup##1{#2}}%`

In addition it defines the alphabet the way it should be used from now on.

549 `\gdef#3{\relax\ifmmode \else \non@alpherr#3\fi`
550 `\use@mathgroup##1{#2}}%`

Finally, we only have to call this macro `\reserved@a` on the old definitions recorded in `\reserved@b`:

551 `\expandafter\reserved@a\reserved@b\@nil`
552 `}`

`\math@bgroup` Here are the default definitions for `\math@bgroup` and `\math@egroup`. We use
`\math@egroup` `\bgroup` instead of `\begingroup` to avoid ‘leaking out’ of style changes. This has the side effect of always producing mathord atoms.

```

553 \let\math@bgroup\bgroup
554 \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`.

```

555 \gdef\calculate@math@sizes{%
556   \@font@info{Calculating\space math\space sizes\space for\space
557     size\space <\f@size>}%
558   \dimen@ \f@size \p@
559   \@tempdimb \defaultscriptratio \dimen@
560   \dimen@ \defaultscriptscriptratio \dimen@
561   \expandafter\xdef\csname S@\f@size\endcsname{%
562     \gdef\noexpand\tf@size{\f@size}%
563     \gdef\noexpand\s@size{\strip@pt\@tempdimb}%
564     \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
565     \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.

```

566 \def\defaultscriptratio{.7}
567 \def\defaultscriptscriptratio{.5}

```

`\noaccents@` If we don't have a definition for `\noaccents@` we provide a dummy.

```

568 \ifx\noaccents@\@undefined
569   \let\noaccents@\@empty
570 \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^AT_EX.

```

571 \if2ekernel
572 \ifx\latexrelease\IncludeInRelease{2017/01/01}\showhyphens{%
573 \ifx\latexrelease\XeTeXsupportfor\showhyphens}%
574 \ifx\XeTeXcharclass\@undefined
575 \ifx\XeTeXcharclass\@undefined
Version for engines other than XeTEX.
576 \DeclareRobustCommand\showhyphens[1]{%
577   \setbox0\vbox{%
578     \color@begingroup
579     \everypar{}%
580     \parfillskip\z@skip\hsize\maxdimen
581     \normalfont
582     \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
583     \color@endgroup}}
584 \else

```

XeT_EX version. When using system fonts XeT_EX reports consecutive runs of characters as a single item in box logging, which means the standard `\showhyphens` does not work. This version typesets the text into a narrow box to force hyphenation and then reconstructs a horizontal list with explicit hyphens to generate the display. Note that the `lmr` OpenType font is forced, this works even if the characters are not in the font as hyphenation is attempted due to the width of the space

and hyphen character. It may generate spurious Missing Character warnings in the log, these are however suppressed from the terminal output by ensuring that `\tracingonline` is locally zero.

```

585 \DeclareRobustCommand\showhyphens[1]{%
586   \setbox0\vbox{%
587     \usefont{TU}{lmr}{m}{n}%
588     \hsize 1sp %
589     \hbadness\@M
590     \hfuzz\maxdimen
591     \tracingonline\z@
592     \everypar={}%
593     \leftskip\z@skip
594     \rightskip\z@skip
595     \parfillskip\z@skip
596     \hyphenpenalty=-\@M
597     \pretolerance\m@ne
598     \interlinepenalty\z@
599     \clubpenalty\z@
600     \widowpenalty\z@
601     \brokenpenalty1127 %
602     \setbox\z@\hbox{%
603       \noindent
604       \hskip\z@skip
605       #1%
606       \par

```

Note here we stop the loop if made no progress, non-removable items may mean that we can not process the whole list (which would be testable as `\lastnodetype=-1`).

```

607   \loop
608   \@tempswafalse
609   \ifnum\lastnodetype=11\unskip\@tempswatrue\fi
610   \ifnum\lastnodetype=12\unkern\@tempswatrue\fi
611   \ifnum\lastnodetype=13 %
612     \count@\lastpenalty
613     \unpenalty\@tempswatrue
614   \fi
615   \ifnum\lastnodetype=\@ne
616     \setbox\tw@\lastbox\@tempswatrue
617     \setbox0\hbox{\unhbox\tw@\unskip\unskip\unpenalty
618                   \ifnum\count@=1127 \else\ \fi
619                   \unhbox0}%
620   \count@\z@
621   \fi
622   \if@tempswa
623   \repeat
624   \hbadness\z@
625   \hsize\maxdimen
626   \showboxdepth\z@
627   \tolerance\m@ne
628   \hyphenpenalty\z@
629   \noindent\unhbox\z@
630 }}
631 \fi

```

```

632 </2ekernel | latexrelease>
633 <latexrelease>\EndIncludeInRelease
634 <latexrelease>\IncludeInRelease{0000/00/00}{\showhyphens}%
635 <latexrelease>{XeTeX support for \showhyphens}%
636 <latexrelease>\gdef\showhyphens#1{%
637 <latexrelease> \setbox0\vbox{%
638 <latexrelease> \color@begingroup
639 <latexrelease> \everypar{}%
640 <latexrelease> \parfillskip\z@skip\hsize\maxdimen
641 <latexrelease> \normalfont
642 <latexrelease> \pretolerance\m@ne\tolerance\m@ne
643 <latexrelease> \hbadness\z@\showboxdepth\z@\ #1%
644 <latexrelease> \color@endgroup}}
645 <latexrelease>\EndIncludeInRelease
646 <*2ekernel>

\addto@hook We need a macro to add tokens to a hook.
647 \long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}

\@vpt
648 \def\@vpt{5}

\@vipt
649 \def\@vipt{6}

\@viipt
650 \def\@viipt{7}

\@viiipt
651 \def\@viiipt{8}

\@ixpt
652 \def\@ixpt{9}

\@xpt
653 \def\@xpt{10}

\@xipt
654 \def\@xipt{10.95}

\@xiipt
655 \def\@xiipt{12}

\@xivpt
656 \def\@xivpt{14.4}

\@xviipt
657 \def\@xviipt{17.28}

\@xxpt
658 \def\@xxpt{20.74}

\@xxvpt
659 \def\@xxvpt{24.88}
660 </2ekernel>

```

File p

lftssbas.dtx

This file contains the main implementation of the ‘low level’ font selection commands. See other parts of the L^AT_EX distribution, or *The L^AT_EX Companion* for higher level documentation of the L^AT_EX ‘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.

28 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.

```
1 (*2ekernel)
2 \def\@nomath#1{\relax\ifmmode
3   \@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_EX from scanning too far in certain situations.

```
4 \gdef\no@alphabet@error#1{\relax \ifmmode
5   \@latex@error{Math\space alphabet\space identifier\space
6     \noexpand#1is\space undefined\space in\space math\space
7     version\space ‘\math@version’}%
8   {Your\space requested\space math\space alphabet\space
9     is\space undefined\space in\space the\space current\space
10    math\space version.^^JCheck\space the\space spelling\space
11    or\space use\space the\space \noexpand\SetMathAlphabet\space
12    command.}
13   \fi}
```

`\new@mathgroup` We also give a new name to `\newfam` and `\fam` to avoid verbal confusion (see the introduction).⁴

`\mathgroup`

```
14 %\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@@n}
15 \let\mathgroup\fam
16 %\let\newfam\new@mathgroup
17 \@onlypreamble\new@mathgroup
```

⁴For 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.

29 Macros for setting up the tables

`\DeclareFontShape` The macro `\DeclareFontShape` takes 6 arguments:

```
18 \def\DeclareFontShape{\begingroup
```

First we restore the catcodes of all characters used in the syntax.

```
19 \nfss@catcodes
```

We use `\expandafter \endgroup` to restore catcode in case something goes wrong with the argument parsing (suggested by Tim Van Zandt)

`\DeclareFontShape`

```
20 \expandafter\endgroup
21 \DeclareFontShape@}
22 \def\DeclareFontShape#1#2#3#4#5#6{%
23 \expandafter\ifx\csname #1+#2\endcsname\relax
24 \@latex@error{Font family ‘#1+#2’ unknown}\@eha
25 \else
26 \expandafter
27 \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
28 \csname #5\endcsname}%
29 \def\reserved@a{#6}%
30 \global
31 \expandafter\let\csname#5\endcsname\expandafter\endcsname
32 \ifx\reserved@a@empty
33 \@empty
34 \else
35 \reserved@a
36 \fi
37 \fi
38 }
```

`\DeclareFixedFont` Define a direct font switch that avoids all overhead.

```
39 \def\DeclareFixedFont#1#2#3#4#5#6{%
40 \begingroup
41 \math@fontsfalse
42 \every@math@size{}%
43 \fontsize{#6}\z@
44 \usefont{#2}{#3}{#4}{#5}%
45 \global\expandafter\let\expandafter#1\the\font
46 \endgroup
47 }
```

`\do@subst@correction`

```
48 \def\do@subst@correction{%
49 \xdef\subst@correction{%
50 \font@name
51 \global\expandafter\font
52 \csname \curr@fontshape/\f@size\endcsname
53 \noexpand\fontname\font
54 \relax}%
```

Calling `\subst@correction` after the current group means calling it after we have loaded the substitution font which is done inside a group.

```
55 \aftergroup\subst@correction
56 }
```

`\DeclareFontFamily`

```
57 \def\DeclareFontFamily#1#2#3{%
```

If we want fast checking for the encoding scheme we can just check for `\T@. .` being defined.

```
58 % \@tempswafalse
59 % \def\reserved@b{#1}%
60 % \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
61 %     \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
62 % \cdp@list
63 % \if@tempswa
64 % \ifundefined{T@#1}%
65 %     {%
66 %         \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
67 %     }%
68 % }
```

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`.

```
69     \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`.

```
70     \global
71     \expandafter\let\csname #1+#2\expandafter\endcsname
72         \ifx \reserved@a\@empty
73             \@empty
74         \else \reserved@a
75         \fi
76     }%
77 }
```

`\cdp@list` We initialize the code page list to be empty.

```
78 \let\cdp@list\@empty
79 \@onlypreamble\cdp@list
```

`\cdp@elt`

```
80 \let\cdp@elt\relax
81 \@onlypreamble\cdp@elt
```

`\DeclareFontEncoding`

```
82 \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.

```
83     \begingroup
84     \nfss@catcodes
85     \expandafter\endgroup
86     \DeclareFontEncoding@}
87 \@onlypreamble\DeclareFontEncoding
```

```

88 \def\DeclareFontEncoding#1#2#3{%
89   \expandafter
90   \ifx\csname T@#1\endcsname\relax
91     \def\cdp@elt{\noexpand\cdp@elt}%
92     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
93                   {\default@family}{\default@series}%
94                   {\default@shape}}%

```

To support encoding dependent commands (like accents) we initialise the command `\<encoding>-cmd` to be `\@changed@cmd`. (See `ltoutenc.dtx` for details.)

```

95   \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
96   \else
97     \font@info{Redeclaring font encoding #1}%
98   \fi
99   \global\@namedef{T@#1}{#2}%
100  \global\@namedef{M@#1}{\default@M#3}%

```

Keep a record of the last encoding being declared:

```

101  \xdef\LastDeclaredEncoding{#1}%
102  }
103 \@onlypreamble\DeclareFontEncoding@

```

`\LastDeclaredEncoding` The last encoding being declared by `\DeclareFontEncoding`.

```

104 \def\LastDeclaredEncoding{}

```

`\DeclareFontSubstitution`

```

105 \def\DeclareFontSubstitution#1#2#3#4{%
106   \expandafter
107   \ifx\csname T@#1\endcsname\relax
108     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
109   \else
110     \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`.

```

111     \edef\reserved@a{#1}%
112     \toks@{}%
113     \def\cdp@elt##1##2##3##4{%
114       \def\reserved@b{##1}%
115       \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.

```

116       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
117     \else

```

If `\reserved@a` and `\reserved@b` differ then we simply copy from the old list to the new.

```

118       \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
119     \fi}%
120     \cdp@list

```

```

121      \xdef\cdp@list{\the\toks@}%
122    \endgroup
123    \global
124    \@namedef{D@#1}{%
125      \def\default@family{#2}%
126      \def\default@series{#3}%
127      \def\default@shape{#4}%
128    }%
129  \fi
130 }
131 \@onlypreamble\DeclareFontSubstitution

```

\DeclareFontEncodingDefaults

```

132 \def\DeclareFontEncodingDefaults#1#2{%
133   \ifx\relax#1\else
134     \ifx\default@T\@empty\else
135       \@font@info{Overwriting encoding scheme text defaults}%
136     \fi
137     \gdef\default@T{#1}%
138   \fi
139   \ifx\relax#2\else
140     \ifx\default@M\@empty\else
141       \@font@info{Overwriting encoding scheme math defaults}%
142     \fi
143     \gdef\default@M{#2}%
144   \fi
145 }
146 \@onlypreamble\DeclareFontEncodingDefaults

```

\default@T

\default@M

```

147 \let\default@T\@empty
148 \let\default@M\@empty

```

\DeclarePreloadSizes

```

149 \def\DeclarePreloadSizes#1#2#3#4#5{%
150   \@ifundefined{T@#1}%
151   {\@latex@error{Encoding scheme ‘#1’ unknown}\@eha}%
152   {%

```

Don't know at the moment what this group here does!

```

153   \begin{group}

```

We define a macro `\reserved@f`⁵ that grabs the next *size* and loads the corresponding font. This is done by delimiting `\reserved@f`'s only argument by the token `,` (comma).

```

154   \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_EXbook*: 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.)

```

155   \if>##1>%

```

⁵We cannot use `\@tempa` since it is needed in `\pickup@font`.

```

156         \let\reserved@f\relax
157     \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.

```

158         \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
159         \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.

```

160         \global\expandafter\let\font@name\relax
161     \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.

```

162     \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.

```

163     \reserved@f#5,,%
164     \endgroup
165 }%
166 }
167 \@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.

```

168 \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.

```

169 \def\DeclareMathSizes{%
170     \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
171     {\@DeclareMathSizes{}}%
172 \@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}.

```

```

173 </2ekernel>
174 <latexrelease>\IncludeInRelease{2015/01/01}{\@DeclareMathSizes}%
175 <latexrelease>           {Arbitrary units in \DeclareMathSizes}%
176 <*2ekernel|latexrelease>
177 \def\@DeclareMathSizes #1#2#3#4#5{%
178     \@defaultunits\dimen@ #2pt\relax\@nnil
179     \if $#3$%
180         \expandafter\let\csname S@strip@pt\dimen@\endcsname\math@fontsfalse
181     \else

```



```

182 \@defaultunits\dimen@ii #3pt\relax\@nnil
183 \@defaultunits\@tempdima #4pt\relax\@nnil
184 \@defaultunits\@tempdimb #5pt\relax\@nnil
185 \toks@{#1}%
186 \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
187 \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
188 \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
189 \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
190 \the\toks@
191 }%
192 \fi
193 }%
194 \</2ekernel\latexrelease>
195 \<latexrelease>\EndIncludeInRelease
196 \<latexrelease>\IncludeInRelease{0000/00/00}{\@DeclareMathSizes}%
197 \<latexrelease> {Arbitrary units in \DeclareMathSizes}%
198 \<latexrelease>\def\@DeclareMathSizes#1#2#3#4#5{%
199 \<latexrelease> \@defaultunits\dimen@#2pt\relax\@nnil
200 \<latexrelease> \if$#3$%
201 \<latexrelease> \expandafter \let
202 \<latexrelease> \csname S@\strip@pt\dimen@\endcsname
203 \<latexrelease> \math@fontsfalse
204 \<latexrelease> \else
205 \<latexrelease> \expandafter \gdef
206 \<latexrelease> \csname S@\strip@pt\dimen@\endcsname
207 \<latexrelease> {\gdef\tf@size{#3}\gdef\sf@size{#4}%
208 \<latexrelease> \gdef\ssf@size{#5}%
209 \<latexrelease> #1%
210 \<latexrelease> }%
211 \<latexrelease> \fi}%
212 \<latexrelease>\EndIncludeInRelease
213 \<*2ekernel>
214 \<\onlypreamble\@DeclareMathSizes>

```

30 Selecting a new font

30.1 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.

```

215 \DeclareRobustCommand\fontencoding[1]{%
216 \expandafter\ifx\csname T@#1\endcsname\relax
217 \<\latexerror{Encoding scheme ‘#1’ unknown}\@eha
218 \else
219 \edef\f@encoding{#1}%
220 \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`.

```

221      \let\enc@update\relax
222      \else

```

If current and new encoding differ we define the macro `\enc@update` to contain all updates necessary at `\selectfont` time.

```

223      \let\enc@update\@enc@update
224      \fi
225      \fi
226 }

```

`\@enc@update`

```

227 \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`.

```

228      \expandafter
229      \let
230      \csname\cf@encoding -cmd\endcsname
231      \@changed@cmd

```

Then we turn the one for the new encoding to `\@current@cmd` (see `ltoutenc.dtx` for further explanations).

```

232      \expandafter
233      \let
234      \csname\f@encoding-cmd\endcsname
235      \@current@cmd

```

We execute the default settings `\default@T`, followed by the one for the new encoding.

```

236      \default@T
237      \csname T@\f@encoding\endcsname

```

Finally we change the default substitution values, disable `\enc@update` and make `\f@encoding` officially the current encoding.

```

238      \csname D@\f@encoding\endcsname
239      \let\enc@update\relax
240      \let\cf@encoding\f@encoding
241 }

```

`\enc@update` The default action in `\selectfont` is to do nothing.

```

242 \let\enc@update\relax

```

`\fontfamily`

```

\fontfamily 243 \DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}

```

`\fontseries` There are now defined later (and differently).

```

\fontseries 244 %\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}

```

```

\fontshape 245 %\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
\fontshape

```

`\usefont` 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.

`\fontencoding` needs to do some setup work so we call that, but instead of calling `\fontfamily`, `\fontseries` and `\fontshape` it earlier versions of this

code did, we now set `\f@family`, etc. directly. If we would call `\fontseries` or `\fontshape` as it was done in the past, they would now interact with the existing series and shape which is not desired if we intend to use an explicit font shape!

```
246 \DeclareRobustCommand\usefont[4]{\fontencoding{#1}%
247   \edef\f@family{#2}%
248   \edef\f@series{#3}%
249   \edef\f@shape{#4}\selectfont
250   \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.

```
251 \DeclareRobustCommand\linespread[1]
252   {\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.

```
253 \DeclareRobustCommand\fontsize[2]
254   {\set@fontsize\baselinestretch{#1}{#2}}
```

`\f@linespread` This macro holds the current internal value for `\baselinestretch`.

```
255 \let\f@family\@empty
256 \let\f@series\@empty
257 \let\f@shape\@empty
258 \let\f@size\@empty
259 \let\f@baselineskip\@empty
260 \let\f@linespread\@empty
```

`\cf@encoding`

```
261 \let\f@encoding\@empty
262 \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.

```
263 \def\@defaultunits{\afterassignment\remove@to@nnil}
```

`\strip@pt` This macro strips the characters `pt` produced by using `\the` on a dimen register.

```
\rem@pt
264 \begingroup
265   \catcode‘P=12
266   \catcode‘T=12
267   \lowercase{
```

```

268 \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
269 \expandafter\endgroup\x
270 \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.

```

271 \DeclareRobustCommand\mathversion[1]
272     {\@nomath\mathversion
273     \expandafter\ifx\csname mv@#1\endcsname\relax
274     \@latex@error{Math version ‘#1’ is not defined}\@eha\else
275     \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@currsz` to an invalid value since this will trigger the setup when the formula starts.

```

276 \gdef\glb@currsz{}%

```

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`.

```

277 \aftergroup\glb@settings
278 \fi}

```

If $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ the use of $\$$ (as the primitive $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ function must be hidden.

Anyway, since we don't have this and won't have it in $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2_{\epsilon}$ 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 279 \let\frozen@everymath\everymath
280 \let\frozen@everydisplay\everydisplay

```

```

\everymath Now we provide now user hooks that will be called in the frozen internals.
\everydisplay 281 \newtoks\everymath
282 \newtoks\everydisplay

```

```

\frozen@everymath Now we define the behaviour of the frozen hooks: first check the math setup then
call the user hook.
283 \frozen@everymath = {\check@mathfonts
284 \the\everymath}

```

`\frozen@everydisplay` Ditto for the display hook.

```

285 \frozen@everydisplay = {\check@mathfonts
286                          \the\everydisplay}
```

`\curr@math@size` This holds locally the current math size.

```

287 \let\curr@math@size\@empty
```

30.2 Macros for loading fonts

`\pickup@font` The macro `\pickup@font` which is used in `\selectfont` is very simple: if the font name is undefined (i.e. not known yet) it calls `\define@newfont` to load it.

```

288 \def\pickup@font{%
289   \expandafter \ifx \font@name \relax
290     \define@newfont
291   \fi}
```

`\split@name` `\pickup@font` assumes that `\font@name` is set but it is sometimes called when `\f@family`, `\f@series`, `\f@shape`, or `\f@size` may have the wrong settings (see, e.g., the definition of `\getanddefine@fonts`). Therefore we need a macro to extract font *family*, *series*, *shape*, and *size* from the font name. To this end we define `\split@name` which takes the font name as a list of characters of `\catcode` 12 (without the backslash at the beginning) delimited by the special control sequence `\@nil`. This is not very complicated: we first ensure that `/` has the right `\catcode`

```

292 {\catcode'\/=12
```

and define `\split@name` so that it will define our private `\f@encoding`, `\f@family`, `\f@series`, `\f@shape`, and `\f@size` macros.

```

293 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}%
294                                     \def\f@family{#2}%
295                                     \def\f@series{#3}%
296                                     \def\f@shape{#4}%
297                                     \def\f@size{#5}}}
```

`\curr@fontshape` Abbreviation which may get removed again for speed.

```

298 \def\curr@fontshape{\f@encoding/\f@family/\f@series/\f@shape}
```

`\define@newfont` Now we can tackle the problem of defining a new font.

```

299 \def\define@newfont{%
```

We have already mentioned that the token list that `\split@name` will get as argument must not start with a backslash. To reach this goal we will set the `\escapechar` to `-1` so that the `\string` primitive will not generate an escape character. To keep this change local we open a group. We use `\begingroup` for this purpose since `\define@newfont` might be called in math mode, and an empty `\bgroup... \egroup` would add an empty Ord atom to the math list and thus affect the spacing.

Also locally redefine `\typeout` so that ‘No file ...fd’ Warnings become Font Info message just sent to the log file.

```

300   \begingroup
301   \let\typeout\@font@info
302   \escapechar\m@ne
```

Then we extract *encoding scheme*, *family*, *series*, *shape*, and *size* from the font name. Note the four `\expandafter`'s so that `\font@name` is expanded first, then `\string`, and finally `\split@name`.

```
303 \expandafter\expandafter\expandafter
304 \split@name\expandafter\string\font@name\@nil
```

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.

```
305 % \expandafter\ifx
306 % \csname\curr@fontshape\endcsname \relax
307 \try@load@fontshape % try always
308 % \fi
309 \expandafter\ifx
310 \csname\curr@fontshape\endcsname \relax
311 \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).

```
312 % \csname\curr@fontshape\endcsname
313 \extract@font\fi
```

We are nearly finished and must only restore the `\escapechar` by closing the group.

```
314 \endgroup}
```

```
315 \def\try@load@fontshape{%
316 \expandafter
317 \ifx\csname \f@encoding+\f@family\endcsname\relax
318 \font@info{Trying to load font information for
319 \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.

```
320 \global\expandafter\let
321 \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).

```
322 \nfss@catcodes
323 \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.

```
324 \edef\reserved@a{%
325 \lowercase{%
326 \noexpand\inputIfFileExists{\f@encoding\f@family.fd}}}%
327 \reserved@a\relax
328 {\@input@\f@encoding\f@family.fd}%
329 \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.

```
330 \def\nfss@catcodes{%
```

We start by making `@` a letter and ignoring all blanks and newlines.

```
331 \makeatletter
332 \catcode'\ 9%
333 \catcode'\^^I9%
334 \catcode'\^^M9%
```

Then we set up `\`, `{`, `}`, `#` and `%` in case an `.fd` file is loaded during a verbatim environment.

```
335 \catcode'\z@
336 \catcode'\{ \@ne
337 \catcode'\}\ @tw@
338 \catcode'\#6%
339 \catcode'\^7%
340 \catcode'\%14%
```

Then we make sure that the important syntax parts have the right `\catcode`.

```
341 \@makeother\<%
342 \@makeother\>%
343 \@makeother\*%
344 \@makeother\.%
345 \@makeother\-%
346 \@makeother\/%
347 \@makeother\[%
348 \@makeother\]%
349 \@makeother\'%
350 \@makeother\'%
351 \@makeother\"%
352 }
```

`\LoadFontDefinitionFile` Load and `.fd` files for some encoding and family (if it exists).

```
353 </2ekernel>
354 <*2ekernel | latexrelease>
355 <latexrelease>\IncludeInRelease{2020/02/02}%
356 <latexrelease>\LoadFontDefinitionFile{Loading .fd files}%
357 \def\LoadFontDefinitionFile#1#2{%
358 \begingroup
359 \edef\f@encoding{#1}%
360 \edef\f@family{#2}%
361 \try@load@fontshape
362 \endgroup
363 }
364 </2ekernel | latexrelease>
```

```

365 \latexrelease\EndIncludeInRelease
366 \latexrelease\IncludeInRelease{0000/00/00}%
367 \latexrelease\LoadFontDefinitionFile\{Loading .fd files}%
368 \latexrelease\let\LoadFontDefinitionFile\@undefined
369 \latexrelease\EndIncludeInRelease
370 \latexrelease\EndIncludeInRelease
371 \*2ekernel

```

DeclareFontFamilySubstitution The idea for this macro is stolen from the `substitutefont` package by Günter Milde, with some modifications and a new name.

Its purpose is to provide characters in a special encoding tht are not available in the current font family to be taken from a different family that is visually compatible (or not if you choose badly). For example, you can match the GFS Didot Greek characters with T_EX Gyre Pagella (Palatino) by specifying

```
\DeclareFontFamilySubstitution{LGR}{qpl}{udidot}
```

This way if you ask for the LGR encoding in for the qpl family you get the characters from the udidot family substituted.

We need to ensure that the macro is defined with `\nfss@catcodes` in force (not quite sure why at the moment to be honest).

```

372 \endkernel
373 \*2ekernel | latexrelease
374 \latexrelease\IncludeInRelease{2020/02/02}%
375 \latexrelease\DeclareFontFamilySubstitution\{Provide family substitution}%
376 \begingroup
377 \nfss@catcodes
378 \gdef\DeclareFontFamilySubstitution#1#2#3{%

```

We only provide a set of silent substitutions. The package also (re)declared the family, but this is incorrect in my eyes and it is better to handle that differently.

Of course the families may still need loading at this point and so we arrange for this. Otherwise we might run into trouble because the necessary `\DeclareFontFamily` has not been seen.

```

379 \LoadFontDefinitionFile{#1}{#2}%
380 \LoadFontDefinitionFile{#1}{#3}%
381 \DeclareFontShape{#1}{#2}{m}{it}{<->ssub * #3/m/it}{}%
382 \DeclareFontShape{#1}{#2}{m}{n}{<->ssub * #3/m/n}{}%
383 \DeclareFontShape{#1}{#2}{m}{sc}{<->ssub * #3/m/sc}{}%
384 \DeclareFontShape{#1}{#2}{m}{sl}{<->ssub * #3/m/sl}{}%

```

These days a few more shapes might be around, so we declare those too. If they don't exist then after the first substitution normal fallbacks will happen.

```

385 \DeclareFontShape{#1}{#2}{m}{sw}{<->ssub * #3/m/sw}{}%
386 \DeclareFontShape{#1}{#2}{m}{scit}{<->ssub * #3/m/scit}{}%
387 \DeclareFontShape{#1}{#2}{m}{scsl}{<->ssub * #3/m/scsl}{}%

```

Same game with b and bx, for other weights you are on your own:

```

388 \DeclareFontShape{#1}{#2}{b}{it}{<->ssub * #3/b/it}{}%
389 \DeclareFontShape{#1}{#2}{b}{n}{<->ssub * #3/b/n}{}%
390 \DeclareFontShape{#1}{#2}{b}{scit}{<->ssub * #3/b/scit}{}%
391 \DeclareFontShape{#1}{#2}{b}{scsl}{<->ssub * #3/b/scsl}{}%
392 \DeclareFontShape{#1}{#2}{b}{sc}{<->ssub * #3/b/sc}{}%
393 \DeclareFontShape{#1}{#2}{b}{sl}{<->ssub * #3/b/sl}{}%

```



```

394 \DeclareFontShape{#1}{#2}{b}{sw}{<->ssub * #3/b/sw}{}%
395 \DeclareFontShape{#1}{#2}{bx}{it}{<->ssub * #3/bx/it}{}%
396 \DeclareFontShape{#1}{#2}{bx}{n}{<->ssub * #3/bx/n}{}%
397 \DeclareFontShape{#1}{#2}{bx}{scit}{<->ssub * #3/bx/scit}{}%
398 \DeclareFontShape{#1}{#2}{bx}{scsl}{<->ssub * #3/bx/scsl}{}%
399 \DeclareFontShape{#1}{#2}{bx}{sc}{<->ssub * #3/bx/sc}{}%
400 \DeclareFontShape{#1}{#2}{bx}{sl}{<->ssub * #3/bx/sl}{}%
401 \DeclareFontShape{#1}{#2}{bx}{sw}{<->ssub * #3/bx/sw}{}%
402 }
403 \endgroup
404 /<2ekernel | latexrelease>
405 <latexrelease>\EndIncludeInRelease
406 <latexrelease>\IncludeInRelease{0000/00/00}%
407 <latexrelease> \{\DeclareFontFamilySubstitution\}{Provide family substitution}%
408 <latexrelease>
409 <latexrelease>\let\DeclareFontFamilySubstitution\@undefined
410 <latexrelease>\EndIncludeInRelease
411 <*2ekernel>

```

`\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!

```

412 /<2ekernel>
413 <*2ekernel | latexrelease>
414 <latexrelease>\IncludeInRelease{2019/10/01}%
415 <latexrelease> \{\DeclareErrorFont\}{No side effects please}%
416 \def\DeclareErrorFont#1#2#3#4#5{%
417     \xdef\error@fontshape{%
418         \noexpand\expandafter\noexpand\split@name\noexpand\string
419         \expandafter\noexpand\cname#1/#2/#3/#4/#5\endcsname
420         \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—and now it did.

```

421 %     \gdef\f@encoding{#1}%
422     \gdef\default@family{#2}%
423     \gdef\default@series{#3}%
424     \gdef\default@shape{#4}%
425 }
426 /<2ekernel | latexrelease>
427 <latexrelease>\EndIncludeInRelease
428 <latexrelease>\IncludeInRelease{0000/00/00}%
429 <latexrelease> \{\DeclareErrorFont\}{No side effects please}%
430 <latexrelease>
431 <latexrelease>\def\DeclareErrorFont#1#2#3#4#5{%
432 <latexrelease> \xdef\error@fontshape{%
433 <latexrelease> \noexpand\expandafter\noexpand\split@name\noexpand\string
434 <latexrelease> \expandafter\noexpand\cname#1/#2/#3/#4/#5\endcsname
435 <latexrelease> \noexpand\@nil}%
436 <latexrelease> \gdef\default@family{#2}%

```

```

437 \latexrelease) \gdef\default@series{#3}%
438 \latexrelease) \gdef\default@shape{#4}%
439 \latexrelease) \global\let\f@family\default@family
440 \latexrelease) \global\let\f@series\default@series
441 \latexrelease) \global\let\f@shape\default@shape
442 \latexrelease) \gdef\f@size{#5}%
443 \latexrelease) \gdef\f@baselineskip{#5pt}%
444 \latexrelease)}
445 \latexrelease)\EndIncludeInRelease
446 (*2ekernel)

447 \@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_EX will go into an infinite loop. But if the defaults are set incorrectly one deserves nothing else!

```

448 \endkernel)
449 \latexrelease)\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
450 \latexrelease) {Font substitution in preamble}%
451 (*2ekernel | latexrelease)
452 \def\wrong@fontshape{%
453   \csname D@f@encoding\endcsname % install defaults if in math
454   \edef\reserved@a{\csname\curr@fontshape\endcsname}%
455   \ifx\last@fontshape\reserved@a
456     \errmessage{Corrupted NFSS tables}%
457     \error@fontshape
458   \else

```

Then we warn the user about the mess and set the shape to its default.

```

459   \let\f@shape\default@shape

```

If the combination is not known, try the default *series*.

```

460   \expandafter\ifx\csname\curr@fontshape\endcsname\relax
461     \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!

```

462   \expandafter
463   \ifx\csname\curr@fontshape\endcsname\relax
464     \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}`.

```

465     \begingroup
466     \try@load@fontshape
467   \endgroup
468   \fi \fi
469 \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 TeX 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.

```
470 \font@warning{Font shape ‘\expandafter\string\reserved@a’
471 \expandafter\@gobble\string\undefined\MessageBreak
472 using ‘\curr@fontshape’ instead\@wrong@font@char}%
473 \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`.

```
474 \gdef\@defaultsubs{%
475 \font@warning{Some font shapes were not available, defaults
476 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.

```
477 \global\expandafter\expandafter\expandafter\let
478 \expandafter\reserved@a
479 \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.

```
480 \xdef\font@name{%
481 \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.

```
482 \pickup@font}
483 /2ekernel | latexrelease)
484 (latexrelease)\EndIncludeInRelease
485 (latexrelease)\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
486 (latexrelease) {Font substitution in preamble}%
487 (latexrelease)\def\wrong@fontshape{%
488 (latexrelease) \csname D@\f@encoding\endcsname
489 (latexrelease) \edef\reserved@a{\csname\curr@fontshape\endcsname}%
490 (latexrelease) \ifx\last@fontshape\reserved@a
491 (latexrelease) \errmessage{Corrupted NFSS tables}%
492 (latexrelease) \error@fontshape
493 (latexrelease) \else
494 (latexrelease) \let\f@shape\default@shape
495 (latexrelease) \expandafter\ifx\csname\curr@fontshape\endcsname\relax
```

```

496 \let\f@series\default@series
497 \expandafter
498 \ifx\csname\curr@fontshape\endcsname\relax
499 \let\f@family\default@family
500 \fi \fi
501 \fi
502 \@font@warning{Font shape
503 ' \expandafter\string\reserved@a'
504 \expandafter\@gobble\string\@undefined
505 \MessageBreak
506 using '\curr@fontshape' instead\@wrong@font@char}%
507 \global\let\last@fontshape\reserved@a
508 \gdef\@defaultsubs{%
509 \@font@warning{Some font shapes were not available,
510 defaults substituted.\@gobbletwo}}%
511 \global\expandafter\expandafter\expandafter\let
512 \expandafter\reserved@a
513 \csname\curr@fontshape\endcsname
514 \xdef\font@name{%
515 \csname\curr@fontshape/\f@size\endcsname}%
516 \pickup@font}
517 \EndIncludeInRelease
518 (*2ekernel)

```

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

```
519 \let\@wrong@font@char\@empty
```

`\@defaultsubs` See above.

```
\@defaultsubs 520 \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_EXbook*, p. 382).

```
521 \def\strip@prefix#1>{\}
```

31 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

```
522 \let\install@mathalphabet\gdef
```

`\math@fonts`

```
523 \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.

```
524 %\def\select@group#1#2#3{\relax\ifmmode
```

We do these things locally using `\begingroup` instead of `\bgroup` to avoid the appearance of an empty Ord atom on the math list.

```
525 % \begingroup
```

We set the math fonts for the *family* in question by calling `\getanddefine@fonts` in the correct environment.

```
526 %      \escapechar\m@ne
527 %      \getanddefine@fonts{\csname c@mv@math@version\endcsname}#3%
```

We globally select the math fonts...

```
528 %      \globaldefs\@ne \math@fonts
```

... and close the group to restore `\globaldefs` and `\escapechar`.

```
529 % \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.

```
530 % \xdef#1{\noexpand\use@mathgroup\noexpand#2%
531 %      {\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.

```
532 %      \expandafter\extract@alph@from@version
533 %          \csname mv@math@version\expandafter\endcsname
534 %      \expandafter{\number\csname c@mv@math@version\endcsname}%
535 %      #1%
536 %      \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.

```
537 %\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.

538 `\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`.

539 `\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`.

540 `\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.

541 `\def\reserved@c####1{\gdef#1{##1####1##3}}%`

Then we execute our auxiliary macro.

542 `\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.

543 `\def\reserved@a\select@group#3##1##2\@nil{%`

This macro can now directly rebuild the math version definition by calling `\reserved@c`:

544 `\reserved@c{%`
545 `\getanddefine@fonts{#2}##2%`
546 `\install@mathalphabet#3{%`
547 `\relax\ifmmode \else \non@alpherr#3\fi`
548 `\use@mathgroup##1{#2}}%`

In addition it defines the alphabet the way it should be used from now on.

549 `\gdef#3{\relax\ifmmode \else \non@alpherr#3\fi`
550 `\use@mathgroup##1{#2}}%`

Finally, we only have to call this macro `\reserved@a` on the old definitions recorded in `\reserved@b`:

551 `\expandafter\reserved@a\reserved@b\@nil`
552 `}`

`\math@bgroup` Here are the default definitions for `\math@bgroup` and `\math@egroup`. We use
`\math@egroup` `\bgroup` instead of `\begingroup` to avoid ‘leaking out’ of style changes. This has the side effect of always producing mathord atoms.

```

553 \let\math@bgroup\bgroup
554 \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`.

```

555 \gdef\calculate@math@sizes{%
556   \@font@info{Calculating\space math\space sizes\space for\space
557     size\space <\f@size>}%
558   \dimen@ \f@size \p@
559   \@tempdimb \defaultscriptratio \dimen@
560   \dimen@ \defaultscriptscriptratio \dimen@
561   \expandafter\xdef\csname S@\f@size\endcsname{%
562     \gdef\noexpand\tf@size{\f@size}%
563     \gdef\noexpand\s@size{\strip@pt\@tempdimb}%
564     \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
565     \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.

```

566 \def\defaultscriptratio{.7}
567 \def\defaultscriptscriptratio{.5}

```

`\noaccents@` If we don't have a definition for `\noaccents@` we provide a dummy.

```

568 \ifx\noaccents@\undefined
569   \let\noaccents@\@empty
570 \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^AT_EX.

```

571 \if2ekernel
572 \ifx\latexrelease\IncludeInRelease{2017/01/01}\showhyphens}%
573 \ifx\latexrelease\XeTeX support for \showhyphens}%
574 \ifx\XeTeXcharclass\undefined
575 \ifx\XeTeXcharclass\undefined
Version for engines other than XeTEX.
576 \DeclareRobustCommand\showhyphens[1]{%
577   \setbox0\vbox{%
578     \color@begingroup
579     \everypar{}%
580     \parfillskip\z@skip\hsize\maxdimen
581     \normalfont
582     \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
583     \color@endgroup}}
584 \else

```

XeT_EX version. When using system fonts XeT_EX reports consecutive runs of characters as a single item in box logging, which means the standard `\showhyphens` does not work. This version typesets the text into a narrow box to force hyphenation and then reconstructs a horizontal list with explicit hyphens to generate the display. Note that the `lmr` OpenType font is forced, this works even if the characters are not in the font as hyphenation is attempted due to the width of the space

and hyphen character. It may generate spurious Missing Character warnings in the log, these are however suppressed from the terminal output by ensuring that `\tracingonline` is locally zero.

```

585 \DeclareRobustCommand\showhyphens[1]{%
586   \setbox0\vbox{%
587     \usefont{TU}{lmr}{m}{n}%
588     \hsize 1sp %
589     \hbadness\@M
590     \hfuzz\maxdimen
591     \tracingonline\z@
592     \everypar={}%
593     \leftskip\z@skip
594     \rightskip\z@skip
595     \parfillskip\z@skip
596     \hyphenpenalty=-\@M
597     \pretolerance\m@ne
598     \interlinepenalty\z@
599     \clubpenalty\z@
600     \widowpenalty\z@
601     \brokenpenalty1127 %
602     \setbox\z@\hbox{%
603       \noindent
604       \hskip\z@skip
605       #1%
606       \par

```

Note here we stop the loop if made no progress, non-removable items may mean that we can not process the whole list (which would be testable as `\lastnodetype=-1`).

```

607   \loop
608   \@tempswafalse
609   \ifnum\lastnodetype=11\unskip\@tempswatrue\fi
610   \ifnum\lastnodetype=12\unkern\@tempswatrue\fi
611   \ifnum\lastnodetype=13 %
612     \count@\lastpenalty
613     \unpenalty\@tempswatrue
614   \fi
615   \ifnum\lastnodetype=\@ne
616     \setbox\tw@\lastbox\@tempswatrue
617     \setbox0\hbox{\unhbox\tw@\unskip\unskip\unpenalty
618                   \ifnum\count@=1127 \else\ \fi
619                   \unhbox0}%
620   \count@\z@
621   \fi
622   \if@tempswa
623   \repeat
624   \hbadness\z@
625   \hsize\maxdimen
626   \showboxdepth\z@
627   \tolerance\m@ne
628   \hyphenpenalty\z@
629   \noindent\unhbox\z@
630 }}
631 \fi

```



```

632 </2ekernel | latexrelease>
633 <latexrelease>\EndIncludeInRelease
634 <latexrelease>\IncludeInRelease{0000/00/00}{\showhyphens}%
635 <latexrelease>                {XeTeX support for \showhyphens}%
636 <latexrelease>\gdef\showhyphens#1{%
637 <latexrelease>  \setbox0\vbox{%
638 <latexrelease>    \color@begingroup
639 <latexrelease>    \everypar{}%
640 <latexrelease>    \parfillskip\z@skip\hsize\maxdimen
641 <latexrelease>    \normalfont
642 <latexrelease>    \pretolerance\m@ne\tolerance\m@ne
643 <latexrelease>    \hbadness\z@\showboxdepth\z@\ #1%
644 <latexrelease>    \color@endgroup}}
645 <latexrelease>\EndIncludeInRelease
646 <*2ekernel>

\addto@hook We need a macro to add tokens to a hook.
647 \long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}

\@vpt
648 \def\@vpt{5}

\@vipt
649 \def\@vipt{6}

\@viipt
650 \def\@viipt{7}

\@viiipt
651 \def\@viiipt{8}

\@ixpt
652 \def\@ixpt{9}

\@xpt
653 \def\@xpt{10}

\@xipt
654 \def\@xipt{10.95}

\@xiipt
655 \def\@xiipt{12}

\@xivpt
656 \def\@xivpt{14.4}

\@xviipt
657 \def\@xviipt{17.28}

\@xxpt
658 \def\@xxpt{20.74}

\@xxvpt
659 \def\@xxvpt{24.88}
660 </2ekernel>

```

File q

ltfsstrc.dtx

32 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^AT_EX will stop.

33 A driver for this document

The next bit of code contains the documentation driver file for T_EX, 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^AT_EX 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)
```

34 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`.⁶

```
16 \input{trace.sty}
```

35 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). `infoshow` 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   }
```

⁶This 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 tracefnt package.}}%
69   }

```

We make `infoshow` the default, which in turn defines `\font@warning` and `\font@info`.

```

70 \ExecuteOptions{infoshow}
71 \ProcessOptions
72 </package>

```

We also need a default definition inside the kernel:

```

73 <*2ekernel>
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>

```

36 Macros common to fam.tex and tracefmt.sty

In the first versions of `tracefmt.dtx` some macros of `fam.dtx`⁷ 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`.

36.1 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{%

```

⁷This 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 ‘undebug’. 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^AT_EX'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 \end{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 \ifx\debug\poptracing
131 \fi

```

`\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 \ifx\tracing\@empty
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 \end{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@currrsize` 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@currrsize \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      \endtrace
196      \fi
197  }
198  \endkernel | 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 \endkernel
```

36.2 Math fonts setup

36.2.1 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@currsz` 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@currsz`.
 - (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).

36.2.2 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 (*2kernel|package)
205 \def\check@mathfonts{%
206   \ifx \glb@currsz \f@size
207   (*trace)
208     \ifnum \tracingfonts>\thr@@
209       \@font@info{*** MATH: no change \f@size\space
210         curr/global (\curr@math@size/\glb@currsz)}\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@currsz)}\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@currsz \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 }

```

36.2.3 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 \mathscr -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 \trace)

```

`\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 (+debug) \pushtracing
276 (+debug) \ifnum\tracingfonts<4 \tracingoff
277 (+debug) \else \tracingon\getanddefine@fonts \fi

```

```

278 (*trace)
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 \trace)

```

We append the current `\tf@size` to `#2` to obtain the font name.⁸ 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 (+debug) \poptracing
295   }
296 \endkernel | package)

```

⁸One 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.

37 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\@nil \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. It's 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 not 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}

```

37.1 Sizefunctions

In the following we define some useful size functions.

`\s@fct@` 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}}

```

`\s@fct@gen` This size function generates the external name from the mandatory argument and 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}

```

`\s@fct@genb` This size function is similar to `gen`, but for fonts where the size is encoded in 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. documentation fixes
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 }

```

`\font@aliasinfo` Sometimes a substitution is only done to map a long font name to a standard shape or series, e.g.,

```
\DeclareFontShape{T1}{Roboto-LF}{b}{it}{<-> alias * Roboto-LF/bold/it}{}
```

Using the `ssub` function in that case will give a strange (and incorrect) warning. As an alternative we therefore offer the size function `alias`. It will still add some info into the `.log` file, but no longer complains that the font shape is not available. It is implemented by grabbing the default warning text and replacing it with a new one.

```
493 </2ekernel>
494 <*2ekernel | latexrelease>
495 <latexrelease>\IncludeInRelease{2020/02/01}%
496 <latexrelease>          {\font@aliasinfo}{alias size function}%
497 \DeclareSizeFunction{alias}{\sub@sfcnt\font@aliasinfo}
498 \def\font@aliasinfo#1{%
499   \font@info{Font\space shape\space '\curr@fontshape'\space
500             aliased\space to\MessageBreak '\mandatory@arg'}%
501 }
502 </2ekernel | latexrelease>
503 <latexrelease>\EndIncludeInRelease
504 <latexrelease>\IncludeInRelease{0000/00/00}%
505 <latexrelease>          {\font@aliasinfo}{alias size function}%
506 <latexrelease>\let\s@fct@alias\@undefined
507 <latexrelease>\let\font@aliasinfo\@undefined
508 <latexrelease>
509 <latexrelease>\EndIncludeInRelease
510 <*2ekernel>
```

`\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.

```
511 \DeclareSizeFunction{subf}{\sub@sfcnt\font@warning}
512 \DeclareSizeFunction{ssubf}{\sub@sfcnt\font@info}

513 \def\sub@sfcnt#1{%
514   #1{Font\space shape\space '\curr@fontshape'\space in\space
515     size\space \f@size\space not\space available\MessageBreak
516     external\space font\space '\mandatory@arg'\space used}%
517   \empty@sfcnt#1%
518 }
```

`\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.

```
519 \DeclareSizeFunction{fixed}{\fixed@sfcnt\font@warning}
520 \DeclareSizeFunction{sfixed}{\fixed@sfcnt\font@info}

521 \def\fixed@sfcnt#1{%
522   \ifx\optional@arg\@empty
523     \let\external@font\mandatory@arg
524   \else
```

```

525     \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
526     \fi
527     #1{External\space font\space '\external@font'\space loaded\space
528         for\space size\MessageBreak
529         <\f@size>}%
530 }
531 </2ekernel>

```


File r

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 s

ltfssdcl.dtx

This file contains the main implementation of the font selection scheme commands. See other parts of the L^AT_EX distribution, or *The L^AT_EX 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.

38 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

```

\version@elt<version1>\version@elt<version2>...
\version@elt<versionn>

```

- the list of all math alphabet identifiers. Here every entry has the form:

```

\group@elt<math group number>
{\{<default family>\}\{<default series>\}\{<default shape>\}}.

```

- 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 $\langle font\ info \rangle$ is either $\backslash reserved@e$ (if the combination is not defined yet) or

```

{\langle family \rangle}{\langle series \rangle}{\langle shape \rangle}

```

$\backslash version@list$ We initialize the version list to be empty.

```

16 \let\version@list=\@empty
17 \@onlypreamble\version@list

```

$\backslash version@elt$

```

18 \let\version@elt\relax
19 \@onlypreamble\version@elt

```

$\backslash new@mathversion$ The macro $\backslash new@mathversion$ 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 $\backslash version@list$. We enclose $\backslash version@list$ in braces since it might be empty (if no *version* is defined yet). But this means that we need a suitable number of $\backslash expandafter$ primitives.

```

21 % \expandafter\in@\expandafter#1\expandafter{\version@list}%
22 % \ifin@

```

If so it prints an error message. The $\backslash next$ macro is used to get rid of the four characters $\backslash mv@$ that would otherwise appear at the begin of the version name in the error message.

```

23 % \latexerror{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 $\backslash version@list$. This is very easy: we define $\backslash version@elt$ (which is the delimiter in $\backslash version@list$) to protect itself and the following token from being expanded and simply redefine $\backslash version@list$.

```

26 % \else
27 % \global\expandafter\newcount\csname c@\expandafter
28 % \gobble\string#1\endcsname
29 % \global\csname 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`
 $\langle\textit{alphabet identifier}\rangle\langle\textit{internal group number}\rangle\langle\textit{default font assignments}\rangle\dots$

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\fontencoding{##1}\def\fontfamily{##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             \csname M@#2\expandafter\endcsname
421             \csname \expandafter\@gobble\string#1\space\endcsname#1}%
422         \version@list
423     \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 \begin@group
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@

```

```

\DeclareMathAccent Could do with more checks like allowing single number in #4 lowercase in #4 etc
573 </2ekernel>
574 <*2ekernel | latexrelease>
575 <latexrelease>\IncludeInRelease{2019/10/01}%
576 <latexrelease>           {DeclareMathAccent}{Make math accents robust}%
577 \def\DeclareMathAccent#1#2#3#4{%
578     \expandafter\in@\csname sym#3\expandafter\endcsname
579     \expandafter{\group@list}%
580     \ifin@
581         \begingroup
582         \count\z@=#4\relax
583         \count\tw@\count\z@
584         \divide\count\z@\sixt@@n
585         \count@\count\z@
586         \multiply\count@\sixt@@n
587         \advance\count\tw@-\count@
588         \if\relax\noexpand#1% is command?
589             \edef\reserved@a{\noexpand\in@
590                 {\expandafter\@gobble\string\mathaccent}
591                 {\expandafter\meaning
592                 \csname\expandafter\@gobble\string#1\space\endcsname}}%
593             \reserved@a
594         \ifin@
595             \expandafter\let
596             \csname\expandafter\@gobble\string#1\space\endcsname
597             \@undefined
598             \expandafter\set@mathaccent
599             \csname sym#3\endcsname#1#2%
600             {\hexnumber@\count\z@}\hexnumber@\count\tw@}%
601             \@font@info{Redefining math accent \string#1}%
602         \else
603             \expandafter\ifx
604             \csname\expandafter\@gobble\string#1\endcsname
605             \relax

```

```

606         \expandafter\set@mathaccent
607         \csname sym#3\endcsname#1#2%
608         {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
609     \else
610         \@latex@error{Command '\string#1' already defined}\@eha
611     \fi
612 \fi
613 \else
614     \@latex@error{Not a command name: '\noexpand#1'}\@eha
615 \fi
616 \endgroup
617 \else
618     \@latex@error{Symbol font '#3' is not defined}\@eha
619 \fi
620 }
621 /2kernel | latexrelease)
622 (latexrelease)\EndIncludeInRelease
623 (latexrelease)\IncludeInRelease{0000/00/00}%
624 (latexrelease)      {DeclareMathAccent}{Make math accents robust}%
625 (latexrelease)\def\DeclareMathAccent#1#2#3#4{%
626 (latexrelease)  \expandafter\in@ \csname sym#3\endcsname
627 (latexrelease)  \expandafter{\group@list}%
628 (latexrelease)  \ifin@
629 (latexrelease)  \begingroup
630 (latexrelease)  \count\z@=#4\relax
631 (latexrelease)  \count\tw@ \count\z@
632 (latexrelease)  \divide\count\z@\sixt@@n
633 (latexrelease)  \count@ \count\z@
634 (latexrelease)  \multiply\count@\sixt@@n
635 (latexrelease)  \advance\count\tw@-\count@
636 (latexrelease)  \if\relax\noexpand#1% is command?
637 (latexrelease)  \edef\reserved@a{\noexpand\in@
638 (latexrelease)  {\expandafter\@gobble\string\mathaccent}{\meaning#1}}%
639 (latexrelease)  \reserved@a
640 (latexrelease)  \ifin@
641 (latexrelease)  \expandafter\set@mathaccent
642 (latexrelease)  \csname sym#3\endcsname#1#2%
643 (latexrelease)  {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
644 (latexrelease)  \@font@info{Redeclaring math accent \string#1}%
645 (latexrelease)  \else
646 (latexrelease)  \expandafter\ifx
647 (latexrelease)  \csname\expandafter\@gobble\string#1\endcsname
648 (latexrelease)  \relax
649 (latexrelease)  \expandafter\set@mathaccent
650 (latexrelease)  \csname sym#3\endcsname#1#2%
651 (latexrelease)  {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
652 (latexrelease)  \else
653 (latexrelease)  \@latex@error{Command '\string#1' already defined}\@eha
654 (latexrelease)  \fi
655 (latexrelease)  \fi
656 (latexrelease)  \else
657 (latexrelease)  \@latex@error{Not a command name: '\noexpand#1'}\@eha
658 (latexrelease)  \fi
659 (latexrelease)  \endgroup

```

```

660 \latexrelease \else
661 \latexrelease \@latexerror{Symbol font '#3' is not defined}\@eha
662 \latexrelease \fi
663 \latexrelease}
664 \latexrelease\EndIncludeInRelease
665 (*2ekernel)

666 \@onlypreamble\DeclareMathAccent

\set@mathaccent
667 \end{kernel}
668 (*2ekernel | latexrelease)
669 \latexrelease\IncludeInRelease{2019/10/01}%
670 \latexrelease {\set@mathaccent}{makemath accents robust}%
671 \def\set@mathaccent#1#2#3#4{%
672   \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}%
673   \MakeRobust#2%
674 }
675 \@onlypreamble\set@mathaccent
676 \end{kernel | latexrelease)
677 \latexrelease\EndIncludeInRelease
678 \latexrelease\IncludeInRelease{0000/00/00}%
679 \latexrelease {\set@mathaccent}{makemath accents robust}%
680 \latexrelease
681 \latexrelease\def\set@mathaccent#1#2#3#4{%
682 \latexrelease \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}}
683 \latexrelease
684 \latexrelease\EndIncludeInRelease
685 (*2ekernel)

\DeclareMathSymbol
686 \def\DeclareMathSymbol#1#2#3#4{%
687   \expandafter\in@\csname sym#3\expandafter\endcsname
688   \expandafter{\group@list}%
689   \ifin@
690     \begingroup
691       \count\z@=#4\relax
692       \count\tw@\count\z@
693       \divide\count\z@\sixt@@n
694       \count@\count\z@
695       \multiply\count@\sixt@@n
696       \advance\count\tw@-\count@
697       \if\relax\noexpand#1% is command?

Store the command name with a space attached inside \reserved@@b in case we
look at a robust definition.
698       \edef\reserved@b{\expandafter\noexpand
699         \csname\expandafter\@gobble\string#1\space\endcsname}%

Test both #1 and #1_ for containing mathchar.
700       \edef\reserved@a
701         {\noexpand\in@\expandafter\@gobble\string\mathchar}%
702         {\meaning#1\expandafter\meaning\reserved@b}}%
703       \reserved@a

```

Drop #1_␣ in case it was defined before.

```

704     \global\expandafter\let\reserved@b\@undefined
705     \ifin@
706         \expandafter\set@mathsymbol
707             \csname sym#3\endcsname#1#2%
708             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
709             \@font@info{Redeclaring math symbol \string#1}%
710     \else
711         \expandafter\ifx
712             \csname\expandafter\@gobble\string#1\endcsname
713             \relax
714         \expandafter\set@mathsymbol
715             \csname sym#3\endcsname#1#2%
716             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
717     \else
718         \@latex@error{Command '\string#1' already defined}\@eha
719     \fi
720 \fi
721 \else
722     \expandafter\set@mathchar
723         \csname sym#3\endcsname#1#2
724         {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
725     \fi
726 \endgroup
727 \else
728     \@latex@error{Symbol font '#3' is not defined}\@eha
729 \fi
730 }
731 \@onlypreamble\DeclareMathSymbol

```

\set@mathchar

```

732 \def\set@mathchar#1#2#3#4{%
733     \global\mathcode'#2="\mathchar@type#3\hexnumber@#1#4\relax}
734 \@onlypreamble\set@mathchar

```

\set@mathsymbol

```

735 \def\set@mathsymbol#1#2#3#4{%
736     \global\mathchardef#2"\mathchar@type#3\hexnumber@#1#4\relax}
737 \@onlypreamble\set@mathsymbol

738 %\def\mathsymbol#1#2#3{%
739 %    \@tempcnta=#3\relax
740 %    \@tempcntb\@tempcnta
741 %    \divide\@tempcnta\sixt@@n
742 %    \count@\@tempcnta
743 %    \multiply\count@\sixt@@n
744 %    \advance\@tempcntb-\count@
745 %    \mathchar"\mathchar@type#1\hexnumber@#2%
746 %        \hexnumber@\@tempcnta\hexnumber@\@tempcntb\relax}
747 %
748 %\def\DeclareMathAlphabetCharacter#1#2#3{%
749 %    \DeclareMathSymbol{#1}7{#2}{#3}

```

\DeclareMathDelimiter

```

750 \def\DeclareMathDelimiter#1{%
751   \if\relax\noexpand#1%
752     \expandafter\@DeclareMathDelimiter
753   \else
754     \expandafter\@xxDeclareMathDelimiter
755   \fi
756   #1}
757 \@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.

```

758 \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!

```

759   \begingroup
760   \let\mathalpha\mathord
761   \ifnum7=\mathchar@type{#2}%
762   \endgroup

```

If this branch is taken we have old syntax (5 arguments).

```

763     \expandafter\@firstofone
764   \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.

```

765     \endgroup
766     \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.

```

767     \expandafter\@firstoftwo
768   \fi
769   {\@xDeclareMathDelimiter#1}{#2}{#3}{#4}}
770 \@onlypreamble\@xxDeclareMathDelimiter

```

`\@DeclareMathDelimiter`

```

771 \</2kernel>
772 \<*2kernel | latexrelease>
773 \<latexrelease>\IncludeInRelease{2019/10/01}%
774 \<latexrelease>          {\@DeclareMathDelimiter}{Make math delimiters robust}%
775 \def\@DeclareMathDelimiter#1#2#3#4#5#6{%
776   \expandafter\in@\csname sym#3\expandafter\endcsname
777   \expandafter{\group@list}%
778   \ifin@
779     \expandafter\in@\csname sym#5\expandafter\endcsname
780     \expandafter{\group@list}%
781   \ifin@
782     \begingroup
783     \count\z@=#4\relax

```

```

784      \count\tw@\count\z@
785      \divide\count\z@\sist@n
786      \count@\count\z@
787      \multiply\count@\sist@n
788      \advance\count\tw@-\count@
789      \edef\reserved@c{\hexnumber@\count\z@}\hexnumber@\count\tw@}%
790      %
791      \count\z@=#6\relax
792      \count\tw@\count\z@
793      \divide\count\z@\sist@n
794      \count@\count\z@
795      \multiply\count@\sist@n
796      \advance\count\tw@-\count@
797      \edef\reserved@d{\hexnumber@\count\z@}\hexnumber@\count\tw@}%
798      %
799      \edef\reserved@a{\noexpand\in@
800        {\expandafter\@gobble\string\delimiter}%
801        {\expandafter\meaning\csname\expandafter\@gobble\string#1\space\endcsname}}%
802      \reserved@a
803      \ifin@
804        \expandafter\let\csname\expandafter\@gobble\string#1\space\endcsname\@undefined
805        \expandafter\set@mathdelimiter
806        \csname sym#3\expandafter\endcsname
807        \csname sym#5\endcsname#1#2%
808        \reserved@c\reserved@d
809        \@font@info{Redeclaring math delimiter \string#1}%
810      \else
811        \expandafter\ifx
812        \csname\expandafter\@gobble\string#1\endcsname
813        \relax
814        \expandafter\set@mathdelimiter
815        \csname sym#3\expandafter\endcsname
816        \csname sym#5\endcsname#1#2%
817        \reserved@c\reserved@d
818      \else
819        \@latex@error{Command '\string#1' already defined}\@eha
820      \fi
821    \fi
822  \endgroup
823 \else
824   \@latex@error{Symbol font '#5' is not defined}\@eha
825 \fi
826 \else
827   \@latex@error{Symbol font '#3' is not defined}\@eha
828 \fi
829 }
830 </2ekernel | latexrelease>
831 <latexrelease>\EndIncludeInRelease
832 <latexrelease>\IncludeInRelease{0000/00/00}%
833 <latexrelease>          {\@DeclareMathDelimiter}{Make math delimiters robust}%
834 <latexrelease>\def\@DeclareMathDelimiter#1#2#3#4#5#6{%
835 <latexrelease>  \expandafter\in@\csname sym#3\expandafter\endcsname
836 <latexrelease>    \expandafter{\group@list}%
837 <latexrelease>  \ifin@

```



```

838 \latexrelease\ \expandafter\in@\csname sym#5\expandafter\endcsname
839 \latexrelease\ \expandafter{\group@list}%
840 \latexrelease\ \ifin@
841 \latexrelease\ \begin@group
842 \latexrelease\ \count\z@=#4\relax
843 \latexrelease\ \count\tw@\count\z@
844 \latexrelease\ \divide\count\z@\sist@@n
845 \latexrelease\ \count@\count\z@
846 \latexrelease\ \multiply\count@\sist@@n
847 \latexrelease\ \advance\count\tw@-\count@
848 \latexrelease\ \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
849 \latexrelease\ %
850 \latexrelease\ \count\z@=#6\relax
851 \latexrelease\ \count\tw@\count\z@
852 \latexrelease\ \divide\count\z@\sist@@n
853 \latexrelease\ \count@\count\z@
854 \latexrelease\ \multiply\count@\sist@@n
855 \latexrelease\ \advance\count\tw@-\count@
856 \latexrelease\ \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
857 \latexrelease\ %
858 \latexrelease\ \edef\reserved@a{\noexpand\in@
859 \latexrelease\ \{\expandafter\@gobble\string\delimiter}\meaning#1}}%
860 \latexrelease\ \reserved@a
861 \latexrelease\ \ifin@
862 \latexrelease\ \expandafter\set@mathdelimiter
863 \latexrelease\ \csname sym#3\expandafter\endcsname
864 \latexrelease\ \csname sym#5\endcsname#1#2%
865 \latexrelease\ \reserved@c\reserved@d
866 \latexrelease\ \@font@info{Redeclaring math delimiter \string#1}%
867 \latexrelease\ \else
868 \latexrelease\ \expandafter\ifx
869 \latexrelease\ \csname\expandafter\@gobble\string#1\endcsname
870 \latexrelease\ \relax
871 \latexrelease\ \expandafter\set@mathdelimiter
872 \latexrelease\ \csname sym#3\expandafter\endcsname
873 \latexrelease\ \csname sym#5\endcsname#1#2%
874 \latexrelease\ \reserved@c\reserved@d
875 \latexrelease\ \else
876 \latexrelease\ \@latex@error{Command ‘\string#1’ already defined}\@eha
877 \latexrelease\ \fi
878 \latexrelease\ \fi
879 \latexrelease\ \endgroup
880 \latexrelease\ \else
881 \latexrelease\ \@latex@error{Symbol font ‘#5’ is not defined}\@eha
882 \latexrelease\ \fi
883 \latexrelease\ \else
884 \latexrelease\ \@latex@error{Symbol font ‘#3’ is not defined}\@eha
885 \latexrelease\ \fi
886 \latexrelease\}
887 \latexrelease\ \EndIncludeInRelease
888 \*2kernel)
889 \@onlypreamble\@DeclareMathDelimiter

```

\@xDeclareMathDelimiter

```

890 \def\xDeclareMathDelimiter#1#2#3#4#5{%
891   \expandafter\in@\csname sym#2\expandafter\endcsname
892   \expandafter{\group@list}%
893   \ifin@
894     \expandafter\in@\csname sym#4\expandafter\endcsname
895     \expandafter{\group@list}%
896     \ifin@
897       \begingroup
898         \count\z@=#3\relax
899         \count\tw@\count\z@
900         \divide\count\z@\sist@%n
901         \count@\count\z@
902         \multiply\count@\sist@%n
903         \advance\count\tw@-\count@
904         \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
905         %
906         \count\z@=#5\relax
907         \count\tw@\count\z@
908         \divide\count\z@\sist@%n
909         \count@\count\z@
910         \multiply\count@\sist@%n
911         \advance\count\tw@-\count@
912         \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
913         \expandafter\set@@mathdelimiter
914         \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
915         \reserved@c\reserved@d
916       \endgroup
917     \else
918       \@latex@error{Symbol font ‘#4’ is not defined}\@eha
919     \fi
920   \else
921     \@latex@error{Symbol font ‘#2’ is not defined}\@eha
922   \fi
923 }
924 \@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)

```

925 </2ekernel>
926 <*2ekernel | latexrelease>
927 <latexrelease>\IncludeInRelease{2019/10/01}%
928 <latexrelease>          {\set@mathdelimiter}{make delimiters robust}%
929 \def\set@mathdelimiter#1#2#3#4#5#6{%
930   \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
931     \hexnumber@#2#6 }%
932   \MakeRobust#3%
933 }
934 \@onlypreamble\set@mathdelimiter
935 </2ekernel | latexrelease>
936 <latexrelease>\EndIncludeInRelease
937 <latexrelease>\IncludeInRelease{0000/00/00}%
938 <latexrelease>          {\set@mathdelimiter}{make delimiters robust}%
939 <latexrelease>

```

```

940 \latexrelease\def\set@mathdelimiter#1#2#3#4#5#6{%
941 \latexrelease\  \xdef#3{\delimiter" \mathchar@type#4\hexnumber@#1#5%
942 \latexrelease\  \hexnumber@#2#6 }}
943 \latexrelease\
944 \latexrelease\EndIncludeInRelease
945 \*2kernel)

```

\set@mathdelimiter

```

946 \def\set@mathdelimiter#1#2#3#4#5{%
947 \global\delcode'#3="\hexnumber@#1#4\hexnumber@#2#5\relax}
948 \onlypreamble\set@mathdelimiter

```

\DeclareMathRadical

```

949 \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!

```

950 \expandafter\ifx
951 \csname\expandafter\@gobble\string#1\endcsname
952 \relax
953 \let#1\radical
954 \fi
955 \edef\reserved@a{\noexpand\in@
956 {\expandafter\@gobble\string\radical}{\meaning#1}}%
957 \reserved@a
958 \ifin@
959 \expandafter\in@\csname sym#2\expandafter\endcsname
960 \expandafter{\group@list}%
961 \ifin@
962 \expandafter\in@\csname sym#4\expandafter\endcsname
963 \expandafter{\group@list}%
964 \ifin@
965 \begingroup
966 \count\z@=#3\relax
967 \count\tw@\count\z@
968 \divide\count\z@\sixt@@n
969 \count@\count\z@
970 \multiply\count@\sixt@@n
971 \advance\count\tw@-\count@
972 \edef\reserved@c{%
973 \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
974 \count\z@=#5\relax
975 \count\tw@\count\z@
976 \divide\count\z@\sixt@@n
977 \count@\count\z@
978 \multiply\count@\sixt@@n
979 \advance\count\tw@-\count@
980 \edef\reserved@d{%
981 \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%

```

Coded inline instead of using \set@mathradical

```

982 % \expandafter\set@mathradical
983 % \csname sym#2\expandafter\endcsname
984 % \csname sym#4\endcsname#1%
985 % \reserved@c\reserved@d

```

```

986         \xdef#1{\radical"\expandafter\hexnumber@
987             \csname sym#2\endcsname\reserved@c
988             \expandafter\hexnumber@
989             \csname sym#4\endcsname\reserved@d
990         \relax}%
991     \endgroup
992 \else
993     \@latex@error{Symbol font ‘#4’ is not defined}\@eha
994 \fi
995 \else
996     \@latex@error{Symbol font ‘#2’ is not defined}\@eha
997 \fi
998 \else
999     \@latex@error{Command ‘\string#1’ already defined}\@eha
1000 \fi
1001 }
1002 \@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

```

1003 \let\mathalpha\relax

```

\mathchar@type

```

1004 \def\mathchar@type#1{%
1005     \ifodd 2#1#1 #1\else                % is this non-negative number?
1006         \ifx#1\mathord 0\else
1007         \ifx#1\mathop 1\else
1008         \ifx#1\mathbin 2\else
1009         \ifx#1\mathrel 3\else
1010         \ifx#1\mathopen 4\else
1011         \ifx#1\mathclose 5\else
1012         \ifx#1\mathpunct 6\else
1013             7%                          % anything else is variable ord
1014         \fi
1015     \fi
1016 \fi
1017 \fi
1018 \fi
1019 \fi
1020 \fi
1021 \fi}
1022 \@onlypreamble\mathchar@type

```

\DeclareSymbolFontAlphabet

```

1023 \def\DeclareSymbolFontAlphabet#1#2{%
1024     \expandafter\DeclareSymbolFontAlphabet@
1025     \csname \expandafter\@gobble\string#1\space\endcsname{#2}#1}
1026 \@onlypreamble\DeclareSymbolFontAlphabet

```

\DeclareSymbolFontAlphabet@

```

1027 \def\DeclareSymbolFontAlphabet@#1#2#3{%

```

We use the switch `\if@tempswa` to decide if we can declare this symbol font alphabet.

```
1028 \if@tempswa true
```

First check if #2 is known to be a symbol font

```
1029 \expandafter\in@\csname sym#2\expandafter\endcsname
1030 \expandafter{\group@list}%
1031 \ifin@
```

Check if #1 is defined as a math alphabet defined via `\DeclareMathAlphabet`:

```
1032 \expandafter\in@\expandafter#1\expandafter{\alpha@list}%
1033 \ifin@
```

If so remove it from the `\alpha@list` and from all math version macros.

```
1034 \@font@info{Redeclaring math alphabet \string#3}%
1035 \toks@{}%
1036 \def\alpha@elt##1##2##3{%
1037 \ifx##1#1\else\addto@hook\toks@{\alpha@elt##1##2##3}\fi}%
1038 \alpha@list
1039 \xdef\alpha@list{\the\toks@}%
```

Now we loop over all versions and remove the math alphabet:

```
1040 \def\version@elt##1{%
1041 \begingroup
1042 \toks@{}%
1043 \def\getanddefine@fonts####1####2{%
1044 \addto@hook\toks@{\getanddefine@fonts####1####2}}%
1045 \def\install@mathalphabet####1####2{%
1046 \ifx####1#1\else
1047 \addto@hook\toks@{\install@mathalphabet
1048 #####1{####2}}\fi}%
1049 ##1%
1050 \xdef##1{\the\toks@}%
1051 \endgroup
1052 }%
1053 \version@list
1054 \else
```

If #3 is not defined as a math alphabet check if it is defined at all:

```
1055 \expandafter\ifx
1056 \csname\expandafter\@gobble\string#1\space\endcsname
1057 \relax
```

If it is undefined, fine otherwise check if it is a math alphabet defined via `\DeclareSymbolFontAlphabet`:

```
1058 \else
1059 \edef\reserved@a{%
1060 \noexpand\in@{\string\use@mathgroup}{\meaning#1}}%
1061 \reserved@a
1062 \ifin@
1063 \@font@info{Redeclaring math alphabet \string#3}%
1064 \else
```

Since the command #3 is defined to be something which is not a math alphabet we have to skip redefining it.

```
1065 \if@tempswa false
```

```

1066         \@latex@error{Command ‘\string#3’ already defined}\@eha
1067     \fi
1068 \fi
1069 \fi
1070 \else

```

Since the symbol font is not known we better skip defining this alphabet.

```

1071     \@tempswafalse
1072     \@latex@error{Unknown symbol font ‘#2’}\@eha
1073 \fi
1074 \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 *<math-settings>* are the one for the encoding that is used in the font shape where *\sym<name>* is pointing to. This means that we have to get it from the information stored in *\group@list*. Thus we loop through that list after defining *\group@elt* in a suitable way.

```

1075     \def\group@elt##1##2{%
1076         \expandafter\ifx\csname sym#2\endcsname##1%
1077         \expandafter\reserved@a\string##2\@nil
1078         \fi}%
1079     \def\reserved@a##1##2/##3\@nil{%
1080         \def\reserved@a{##2}}%
1081     \group@list
1082     \toks@{\relax\ifmmode \else \non@alpherr#1\fi}%
1083     \edef#1{\the\toks@
1084         \noexpand\use@mathgroup
1085         \expandafter\noexpand\csname M@\reserved@a\endcsname
1086         \csname sym#2\endcsname}%
1087     \def#3{\protect#1}%
1088 \fi
1089 }
1090 \@onlypreamble\DeclareSymbolFontAlphabet@
1091 /2ekernel)

```

File t

ltfssini.dtx

This file contains the top level L^AT_EX interface to the font selection scheme commands. See other parts of the L^AT_EX distribution, or *The L^AT_EX Companion* for higher level documentation of these commands.

39 NFSS Initialisation

Finally, there are six commands that are to be used in L^AT_EX and that we will therefore protect against expansion at the wrong point: `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize`, `\selectfont`, and `\mathversion`.

```
1 \<2kernel>
```

39.1 Providing math *versions*

L^AT_EX 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.

(Actually most are now defined further down in the file.)

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\sfddefault\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}

```

40 Custom series settings for main document families

This section was introduced 2020/02/02 and for now we support a full rollback (may need splitting later).

```

31 \endkernel
32 \if*2ekernel|latexrelease
33 (latexrelease)\IncludeInRelease{2020/02/02}%
34 (latexrelease)          {\DeclareFontSeriesDefault}{Custom series}%

```

One problem with the NFSS approach of handling the series axis turned out to be that (especially with respect to “boldness”) different font families implemented different strategies. For example, with Computer Modern fonts you normally only have **bx** whereas most PostScript fonts offered only **b** but not **bx**. As a result L^AT_EX’s standard setting for `\bfdefault` didn’t work with such fonts, but if it got changed to produce **b**, then that didn’t work with Computer Modern if the fonts got combined (e.g., using Computer Modern Typewriter with such fonts).

The solution back then was to provide substitution rules in the font `.fd` such that if a **bx** series got requested the **b** series got used. While this works in that particular case, it isn’t a very general solution. For example, if you happen to have a font family that has several weights you may want to typeset the whole document in a somewhat lighter or darker font but if you then modify `\mddefault` to allow for this, then of course your change only works with that particular family but not with the typewriter or sans serif family you also want to use.

A better solution was provided by the `mweights` package by Bob Tennent that offers defaults on the level of the three main font families in the document: for “rm”, “sf” and “tt” so that font packages could define defaults for the sans serif document font by providing `\bfseries@sf` which then was used when `\bfseries` got executed and the current family was the `\sffamily`.

`\DeclareFontSeriesDefault`

We now support this concept directly from within L^AT_EX and for use in font packages (or the document preamble) we offer `\DeclareSeriesDefault`. This declaration takes three arguments:

document family interface: Can either be `rm`, `sf` or `tt`. This is optional and if not given the overall default.

document series interface: Can be `md` or `bf`.

series value: This is the value that is going to be used with the combination is requested.

For example, `\DeclareFontSeriesDefault[rm]{bf}{sb}` would use **sb** (semi-bold) when `\rmfamily \bfseries` is asked for.

If used without the optional argument, e.g., `\DeclareFontSeriesDefault{bf}{b}` then this is like redefining `\bfdefault` or `\mddefault`.

If some family specific defaults aren't given, e.g. if there are no declarations for, say, `tt` then the format defaults of `\mddefault` and `\bfdefault` are assumed. If those are later changed this is *not* reflected!⁹

`\DeclareFontSeriesDefault` The command to declare font series defaults for the “rm”, “sf” or “tt” family.

```
35 \newcommand\DeclareFontSeriesDefault[3][]{%
36   \def\@resereda{#1}%
```

No optional argument: set up general default.

```
37   \ifx\@resereda\@empty
38     \ifcsname #2series\endcsname           % supported are
39                                           % \[md/bf]default
40     \expandafter\def
41       \csname #2default\endcsname{#3}%
42   \else
43     \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
44     {Mandatory first argument must be 'md' or 'bf'.}
45   \fi
```

Optional argument given, set up specific default.

```
46   \else
47     \ifcsname #2series@#1\endcsname         % supported are
48                                           % \[md/bf]series@[rm/sf/tt]
49     \expandafter\edef
50       \csname #2series@#1\endcsname{#3}%
```

If the interface is used we remove the frozen kernel default. This way, we know that something was explicitly set up (even if the setup has the same value as the default).

```
51     \expandafter\let
52       \csname #2series@#1@kernel\endcsname\@undefined
53   \else
54     \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
55     {Optional first argument must be 'rm', 'sf', or 'tt'. \MessageBreak
56     Second argument must be 'md' or 'bf'.}
57   \fi
58 \fi
59 }
```

`\mdseries@rm` We initialize the family specific default at the end of the format generation. Later on they may get overwritten in the preamble or a package via `\DeclareFontSeriesDefault` (or possibly directly).

`\mdseries@sf` Conceptual change: The `\bfdefault` will be `b` not `bx` because that is what it should be really for nearly every font except Computer/Latin Modern.

`\mdseries@tt` To account for the fact that by default we typeset in CM or LM we set up the `\bfseries@..` defaults to use `bx` instead.

This means that it behaves like before because if the default fonts are used then `\bfseries@rm` etc kick in and make `\textbf` use `bx`. However, if the font gets changed then `\bfdefault` will get used.

```
60 \def\bfseries@rm{bx}
```

⁹I see no easy way to achieve this without compromising compatibility with existing packages that currently use `mweights` and directly define (some) of the `\mdseries@..` commands but not others.

```

61 \def\bfseries@sf{bx}
62 \def\bfseries@tt{bx}

```

Frozen version of the kernel defaults so we can see if they have changed.

```

63 \let\bfseries@rm@kernel\bfseries@rm
64 \let\bfseries@sf@kernel\bfseries@sf
65 \let\bfseries@tt@kernel\bfseries@tt

```

The default for the medium series is `m` and this will be interpreted as resetting both weight and width. To reset only one of them the virtual value `?m` and `m?` are available.

```

66 \def\mdseries@rm{m}
67 \def\mdseries@sf{m}
68 \def\mdseries@tt{m}

```

`\expand@font@defaults` The family specific defaults are fully expanded, i.e., they are defined via `\edef` inside `\DeclareFontSeriesDefault`. However, the overall defaults, e.g., `\bfdefault` may have been redefined by the user and thus may not be fully expanded. So to enable reliable comparison we make expanded versions of them. That we rerun each time. The alternative would be to only allow for changes before begin document.

```

69 \def\expand@font@defaults{%
70   \edef\rmdef@ult{\rmdefault}%
71   \edef\sdef@ult{\sfdefault}%
72   \edef\ttdef@ult{\ttdefault}%
73   \edef\bfdef@ult{\bfdefault}%
74   \edef\mddef@ult{\mddefault}%
75   \edef\famdef@ult{\familydefault}%
76 }

```

`\bfseries` This document command switches to the bold series.

```

77 \DeclareRobustCommand\bfseries{%
78   \not@math@alphabet\bfseries\mathbf

```

In the original NFSS definition it then called `\fontseries` with the value `\bfdefault`. In the new scheme we have more alternatives and therefore check if the current family (`\f@family`) is the current `\rmdef@ult`, `\sfdef@ult` or `\ttdef@ult` and then select the correct family default in that case.

```

79   \expand@font@defaults
80   \fontseries{%
81     \ifx\f@family\rmdef@ult      \bfseries@rm
82     \else\ifx\f@family\sdef@ult  \bfseries@sf
83     \else\ifx\f@family\ttdef@ult \bfseries@tt

```

If not `\bfdefault` is used.

```

84     \else                        \bfdefault
85     \fi\fi\fi
86   }%
87   \selectfont
88 }

```

`\mdseries` This document command switches to the medium series.

```

89 \DeclareRobustCommand\mdseries{%
90   \not@math@alphabet\mdseries\relax

```

```

91 \expand@font@defaults
92 \fontseries{%
93   \ifx\f@family\rmdef@ult      \mdseries@rm
94   \else\ifx\f@family\sfdef@ult \mdseries@sf
95   \else\ifx\f@family\ttdef@ult \mdseries@tt
96   \else                        \mddefault
97   \fi\fi\fi
98 }%
99 \selectfont
100 }

```

`\rmfamily` Here are the document level commands for changing the main font families, or rather, here is a documented outline of the code, the actual code is then streamlined and somewhat generalized.

```

\DeclareRobustCommand\rmfamily{%
  \not@math@alphabet\rmfamily\mathrm

```

If families are changed then we have to do a bit more work. In the original NFSS implementation a family change kept encoding, series shape and size unchanged but now we can't any longer simply reuse the current series value. Instead we may have to change it from one family default to the next.

```
\expand@font@defaults
```

We have to do the testing while the current family is still unchanged but we have to do the adjustment of the series after it got changed (because the new family might has different sets of shapes available and we certainly don't want to see substitution going on. So we use `\target@series@value` to hold the target series (if any).

```
\let\target@series@value\@empty
```

Thus, if the current family is the sans family

```
\ifx\f@family\sfdef@ult
```

and if we using the medium series of the sans family

```
\ifx\f@series\mdseries@sf
```

then lets switch to the medium series for the serif family

```
\let\target@series@value\mdseries@rm
```

and if we use the bold series of the sans family switch to the bold default of the serif family:

```
\else\ifx\f@series\bfseries@sf \let\target@series@value\bfseries@rm
```

However, the sans family may not have any specific defaults set, so we also compare with the overall defaults.

```

\else\ifx\f@series\mddef@ult \let\target@series@value\mdseries@rm
\else\ifx\f@series\bfdef@ult \let\target@series@value\bfseries@rm

```

If neither test was true we leave the series alone. This way a special manual setting such as `\fontseries{lc}` is not undone if the family changes (of course there may not be any support for it in the new family but then the NFSS substitution kicks in and sorts it out).

```
\fi\fi\fi\fi
```

We need to do the same if the current family is the typewriter family:

```
\else\ifx\f@family\ttdef@ult
  \ifx\f@series\mdseries@tt      \let\target@series@value\mdseries@rm
  \else\ifx\f@series\bfseries@tt \let\target@series@value\bfseries@rm
  \else\ifx\f@series\mddef@ult   \let\target@series@value\mdseries@rm
  \else\ifx\f@series\bfdef@ult   \let\target@series@value\bfseries@rm
  \fi\fi\fi\fi
\fi\fi
```

With these preparations for series out of the way we can now change the font family to `\rmdefault`.

```
\fontfamily\rmdefault
```

If `\target@series@value` is still empty there is nothing more to do other than selecting the new family. However, if not then we should update the font series now as well. But there is one further subtle issue. We may not have loaded an `.fd` file for our target font family yet. In the past that was done in `\selectfont` if necessary but since we are now doing all the comparisons in `\fontseries` we need to make sure that the font family specifications are already loaded prior to calling `\fontseries`.

```
\ifx\target@series@value\@empty \else
  \maybe@load@fontshape
```

Updating the series in this case means directly changing `\f@series` to the target value. We don't want to go through `\fontseries` because that would apply the mappings and then `bx + b` would keep `bx` instead of changing to `b` as desired. as

```
\let\f@series\target@series@value
\fi
\selectfont}
```

So now for the real definition: most of the code above gets delegated to a helper command `\prepare@family@series@update` so that the definition becomes again fairly short. In addition we add a hook, mainly for our Japanese friends so that the code can be extended prior to the call to `\selectfont`.

```
101 \DeclareRobustCommand\rmfamily{%
102   \not@math@alphabet\rmfamily\mathrm
```

This holds all the code discussed above, first argument is the meta family, i.e., `rm` in this case, and second argument is the default family name, e.g., `cmr` indirectly accessed via `\rmdefault`. This is calling `\fontfamily` and if necessary `\fontseries` as outline above.

```
103   \prepare@family@series@update{rm}\rmdefault
```

The comes the hook code (by default a no-op) and finally the call to `\selectfont`.

```
104 \rmfamilyhook
105 \selectfont}
```

`\sffamily` The definitions for `\sffamily` and `\ttfamily` are similar, the differences are only
`\ttfamily` in what font families get checked.

```
106 \DeclareRobustCommand\sffamily{%
107 \not@math@alphabet\sffamily\mathsf
108 \prepare@family@series@update{sf}\sfdefault
109 \@sffamilyhook
110 \selectfont}

111 \DeclareRobustCommand\ttfamily{%
112 \not@math@alphabet\ttfamily\mathtt
113 \prepare@family@series@update{tt}\ttdefault
114 \@ttfamilyhook
115 \selectfont}
```

`\rmfamilyhook` By default the hooks do nothing.

`\@sffamilyhook` 116 `\let\rmfamilyhook\@empty`

`\@ttfamilyhook` 117 `\let\@sffamilyhook\@empty`

118 `\let\@ttfamilyhook\@empty`

`\prepare@family@series@update` This is core command that prepares for the family update. The big difference to the documented code above is that the nested `\ifx` statements seem to be missing. Instead we loop through an internal list that holds the names of the three meta families. This approach allows us to extend the mechanism at a later stage to allow for additional named meta families.

`\@meta@family@list` Here is the current definition of that list:

```
119 \def\@meta@family@list{\@elt{rm}\@elt{sf}\@elt{tt}}

120 \def\prepare@family@series@update#1#2{%
121 \expand@font@defaults
```

We prepare for changing the current series. We have to find it before changing the family as discussed above.

```
122 \let\target@series@value\@empty
123 \def\target@meta@family@value{#1}%
```

To find it we loop over the meta family list with a suitable definition of `\@elt`.

```
124 \let\@elt@update@series@target@value
125 \@meta@family@list
126 \let\@elt\relax
```

That will figure out the correct series value to use without updating it. Now we can change the family.

```
127 \fontfamily#2%
```

After that we update the series. That code is again like the one above.

```
128 \ifx\target@series@value\@empty
129 \else
130 \maybe@load@fontshape
131 \let\f@series\target@series@value
132 \fi
133 }
```

`\update@series@target@value` In this macro used in the look you basically find the nested `\ifxs` from the outline above. The only difference is that it is parameterized instead of being written out and only for one block of tests because the code is called repeatedly when looping over the meta family list. From the list we get each meta family name in turn.

```

134 \def\update@series@target@value#1{%
    There is one additional test at the beginning, because the list contains all meta
    families and we need to ignore the case where current one from the list and target
    one are identical.
    135 \def\@reserveda{#1}%
    136 \ifx\target@meta@family@value\@reserveda % rm -> rm do nothing
    137 \else
    We only “do” something if the current font family matches the current meta family.
    138 \expandafter\ifx\csname#1def@ult\endcsname\family
    If that’s the case we know that this is the block that applies (only one meta family
    can match). So to speed things up we change \@elt so that the rest of the loop
    gets gobbled.
    139 \let\@elt\@gobble
    Then we try to find the right new value for the series (as explained above). The
    two macros defined first are only there because we now need to use \csname and
    this way the code will be a little faster.
    140 \expandafter\let\expandafter\@reservedb
    141 \csname mdseries@\target@meta@family@value\endcsname
    142 \expandafter\let\expandafter\@reservedc
    143 \csname bfseries@\target@meta@family@value\endcsname
    This here is now identical to the nested \ifx block from the outline, except that it
    there appeared twice in \rmfamily. This is now covered by looping and stopping
    the loop when a match was found.
    144 \expandafter\ifx\csname mdseries@#1\endcsname\familyseries
    145 \let\target@series@value\@reservedb
    146 \else\expandafter\ifx\csname bfseries@#1\endcsname\familyseries
    147 \let\target@series@value\@reservedc
    148 \else\ifx\familyseries\mddef@ult \let\target@series@value\@reservedb
    149 \else\ifx\familyseries\bfdef@ult \let\target@series@value\@reservedc
    150 \fi\fi\fi\fi
    151 \fi
    152 \fi
    153 }

```

`\init@series@setup` This is code to be run at begin document ...

```

154 \def\init@series@setup{%

```

We only want `bx` in `\bfseries@rm` if the roman font is Computer Modern or Latin Modern, otherwise it should be `b`. It was set to `bx` in the kernel so that any font use with the default families in the preamble get this value. Now at the real document start we check if the fonts have been changed. If there was a `\DeclareFontSeriesDefault` declaration or `\bfseries@rm` was directly altered then it differs from `\bfseries@rm@kernel` and we do nothing. Otherwise we check if `\rmdefault` is one of the CM/LM font families and if so we keep `bx` otherwise we change it to `b`.

This approach doesn't cover one case: CM/LM got changed to a different family that supports `bx`, but the support package for that family used `\def\bfseries@rm{bx}` instead of using `\DeclareFontSeriesDefault`. In that case the code here changes it to `b`. Solution: use the `\DeclareFontSeriesDefault` interface.

```
155 \ifx\bfseries@rm@kernel\bfseries@rm
156   \expandafter\in@\expandafter{\rmdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
157   \ifin@ \else \def\bfseries@rm{b}\fi\fi
```

Same approach for `\bfseries@sf` and `\bfseries@tt`:

```
158 \ifx\bfseries@sf@kernel\bfseries@sf
159   \expandafter\in@\expandafter{\sfdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
160   \ifin@ \else \def\bfseries@sf{b}\fi\fi
161 \ifx\bfseries@tt@kernel\bfseries@tt
162   \expandafter\in@\expandafter{\ttdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
163   \ifin@ \else \def\bfseries@tt{b}\fi\fi
```

If the document preamble has changed the `\familydefault` or if the if the `\rmdefault` contains a new font family, we have to adjust the series defaults accordingly, before starting typesetting.

On the other hand if we still typeset in CM or LM then `\bfdefault` is wrong since it is now saying `b` and not `bx`.

To fix this we run `\rmfamily`, `\sffamily` or `\ttfamily` depending on the situation and this will correct the setup for us.

```
164 \expand@font@defaults
165 \ifx\famdef@ult\rmdef@ult \rmfamily
166 \else\ifx\famdef@ult\sfdef@ult \sffamily
167 \else\ifx\famdef@ult\ttdef@ult \ttfamily
168 \fi\fi\fi
169 }%
```

As the kernel code now implements the same functionality as `mweights`, albeit internally coded slightly differently, that package shouldn't be loaded any more. We therefore pretend that it already got loaded. Thus, a font package that tries to load it and then sets `\mdseries@...`, etc. will continue to work but will now use the kernel code.

Of course, mid-term such package should probably use `\DeclareFontSeriesDefault` instead of making using low-level definitions.

```
170 \expandafter\let\csname ver@mweights.sty\endcsname\fmtversion
171 </2kernel | latexrelease>
172 <latexrelease>\EndIncludeInRelease
173 <latexrelease>\IncludeInRelease{0000/00/00}%
174 <latexrelease>          {\DeclareFontSeriesDefault}{Custom series}%
175 <latexrelease>
176 <latexrelease>\let\DeclareFontSeriesDefault\@undefined
177 <latexrelease>\let\bfseries@rm\@undefined
178 <latexrelease>\let\bfseries@sf\@undefined
179 <latexrelease>\let\bfseries@tt\@undefined
180 <latexrelease>\let\bfseries@rm@kernel\@undefined
181 <latexrelease>\let\bfseries@sf@kernel\@undefined
182 <latexrelease>\let\bfseries@tt@kernel\@undefined
183 <latexrelease>\let\mdseries@rm\@undefined
```

```

184 \let\mdseries@sf\@undefined
185 \let\mdseries@tt\@undefined
186 \let\expand@font@defaults\@undefined
187 \expandafter\let\csname ver@weights.sty\endcsname\@undefined
188 \let\@fontseries\@empty
189 \DeclareRobustCommand\bfseries
190 \let\bfseries\fontseries\bfdefault\selectfont
191 \let\fontseries\bfdefault\selectfont
192 \DeclareRobustCommand\mdseries
193 \let\mdseries\fontseries\mddefault\selectfont
194 \let\fontseries\mddefault\selectfont
195 \DeclareRobustCommand\rmfamily
196 \let\rmfamily\fontfamily\rmdefault\selectfont
197 \let\fontfamily\rmdefault\selectfont
198 \DeclareRobustCommand\sffamily
199 \let\sffamily\fontfamily\sfdefault\selectfont
200 \let\fontfamily\sfdefault\selectfont
201 \DeclareRobustCommand\ttfamily
202 \let\ttfamily\fontfamily\ttdefault\selectfont
203 \let\fontfamily\ttdefault\selectfont
204 \let\fontfamily\@empty
205 \let\@rmfamilyhook\@undefined
206 \let\@sffamilyhook\@undefined
207 \let\@ttfamilyhook\@undefined
208 \let\@metafamilylist\@undefined
209 \let\prepare@family@series@update\@undefined
210 \let@update@series@target@value\@undefined
211 \let\@fontseries\@empty

```

This is always called in `\document` so don't make it undefined.

```

212 \let\init@series@setup\relax
213 \let\@fontseries\@empty
214 \EndIncludeInRelease
215 \*2kernel

```

41 Supporting nested emphasis

By default $\text{\LaTeX} 2_{\epsilon}$ supports two levels of nested emphasis: if the current font has an upright shape then it switches to `\itshape` otherwise to `\emminnershape` (which defaults to `\upshape`). This means nested emphasis will oscillate between italic and upright shapes.

Sometimes it would be nice to allow for a more lengthy sequence, but instead of providing a fixed one \LaTeX now offers a general mechanism that allows to define arbitrary sequences.

`\DeclareEmphSequence`
`\emforce`

This declaration expects a comma separated list of (font) change declarations corresponding to increasing levels of emphasis. The mechanism tries to be “smart” and verifies that the declarations actually alter the font. If not it will ignore this level and tries the next one—the assumption being that there was a manual font change in the document to the font that is now supposed to be used for emphasis. Of course, this only works if the declarations in the list actually change the font

and not, say, just the color. In such a case one has to use `\emforce` to which directs the mechanism to use the level even if the font attributes haven't changed.

`\emreset` If the nesting is so deep, that the specified levels are exhausted then `\emreset` is used as a final set of declarations (which by default returns back to the upright shape). Any additional nesting levels will then reuse the list from its beginning.

`\DeclareEmphSequence` `\DeclareEmphSequence` expects a list of declaration. Spaces in the argument are dropped to avoid surious spaces in the output. The declarations are additive. At the very end the shape is reset using `\emreset` and `\emforce` so that this case is never skipped.¹⁰ Further nested calls restart at the beginning.

```

216 \endkernel
217 (*2kernel | latexrelease)
218 (latexrelease) \IncludeInRelease{2020/02/02}%
219 (latexrelease) { \DeclareEmphSequence}{Nested emph}%
220 \def\DeclareEmphSequence#1{%
221   \protected@edef\emfontdeclare@clist{\zap@space#1, \@empty\emforce\emreset}%
222 }
```

By default the it is empty, in which case `\emminnershape` is used by L^AT_EX.

```
223 \let\emfontdeclare@clist\@empty
```

`\emrest` Reset the font to upright and upper/lower case. With the default rules using `\shapedefault` does that for us but to be on the safe side we do it like this:

```
224 \DeclareRobustCommand\emreset{\upshape\ulcshape}
```

`\em` The new definition for `\em` (and implicitly `\emph` is like it was before if `\emfontdeclare@clist` is empty.

```

225 \DeclareRobustCommand\em{%
226   \@nomath\em
227   \ifx\emfontdeclare@clist\@empty
228     \ifdim \fontdimen\@ne\font >\z@
229       \emminnershape \else \itshape \fi
230   \else
```

But if not we use the list to decide how to do emphasis.

We use the current font to check if the declarations have any effect, so even a size change is allowed and identified as a modification (but a color change, for example, isn't). So first we save the current status.

```
231 \edef\em@currfont{\csname\curr@fontshape/\f@size\endcsname}%
```

Then we grab the next element from the list and check if it can be used.

```

232 \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
233 \fi
234 }
235 \def\emminnershape{\upshape}
```

`\do@emfont@update` We know that the list (if not empty) has at least 2 elements separated by a comma, so we pick up the first in #1 and the rest in #2.

```
236 \def\do@emfont@update#1,#2\do@emfont@update{%
```

¹⁰Maybe we should not add `\emforce` but allow that case to be skipped as well. Of course, that might result in an endless loop if somebody defines a sequence without any font change and without `\emforce` but ...


```

267 \latexrelease)      {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
268 \latexrelease)      \upshape \else \itshape \fi}%
269 \latexrelease)\let\eminnershape\@undefined
270 \latexrelease)\EndIncludeInRelease
271 (*2kernel)

```

`\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`.

```

272 \def\not@math@alphabet#1#2{%
273   \relax
274   \ifmmode
275     \@latex@error{Command \noexpand#1invalid in math mode}%
276     {%
277       Please
278       \ifx#2\relax
279         define a new math alphabet^^J%
280         if you want to use a special font in math mode%
281       \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 :-).

```

282         use the math alphabet \noexpand#2instead of
283         the #1command%
284       \fi
285       .
286     }%
287   \fi}

```

Finally we provide two abbreviations to switch to the *L^AT_EX versions*.

```

288 \DeclareRobustCommand\boldmath{\@nomath\boldmath
289   \mathversion{bold}}
290 \DeclareRobustCommand\unboldmath{\@nomath\unboldmath
291   \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`.

```

292 \def\math@version{normal}

```

41.1 Miscellaneous

`\newfont` We start by defining a few macros that are part of standard *L^AT_EX*'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.

```

293 \def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
294 \DeclareRobustCommand\symbols[1]{\char #1\relax}

```

`\@setfontsize` This abbreviation is used by *L^AT_EX*'s user level size changing commands, such as
`\@setsize` `\large`.

```

295 \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`.

```
296 \ifx\protect\@typeset@protect
297 \let\@currsiz#1%
298 \fi
299 \fontsize{#2}{#3}\selectfont}
```

For compatibility we also define `\@setsize` the 209 command

```
300 \*compat
301 \def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
302 \end{compat}
```

`\hexnumber@` To set up L^AT_EX's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple `\ifcase`.

```
303 \def\hexnumber#1{\ifcase\number#1
304 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
305 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.

```
306 \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`.

```
307 %\DeclareRobustCommand\copyright
308 % \{\oalign{\hfil
309 % \raise.07ex\hbox{\mdseries\upshape c}\hfil\cr
310 % \mathhexbox20D}}}
```

`\normalfont` The macro `\reset@font` is used in L^AT_EX 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 L^AT_EX version above but nevertheless are able to run all kind of mixtures.

The user interface name for `\reset@font` is `\normalfont`:

```
311 \DeclareRobustCommand\normalfont
312 \{\usefont\encodingdefault
313 \familydefault
314 \seriesdefault
315 \shapedefault
316 \relax}
317 \let\reset@font\normalfont
```

We left out the special L^AT_EX 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.

```

318 \def\not@base#1{\@latex@error
319   {Command \noexpand#1not provided in base LaTeX2e}%
320   {Load the latexsym or the amsfonts package to
321     define this symbol}}
322 \def\mho{\not@base\mho}
323 \def\Join{\not@base\Join}
324 \def\Box{\not@base\Box}
325 \def\Diamond{\not@base\Diamond}
326 \def\leadsto{\not@base\leadsto}
327 \def\squsubset{\not@base\squsubset}
328 \def\sqsupset{\not@base\sqsupset}
329 \def\lhd{\not@base\lhd}
330 \def\unlhd{\not@base\unlhd}
331 \def\rhd{\not@base\rhd}
332 \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.

```

333 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}  %% don't modify this setting
334                                         %% overwrite it in fontdef.cfg
335                                         %% if necessary

```

We also set some default values for `\f@family` etc. Note that we don't yet have any encodings that comes later. In the past this was implicitly done by `\DeclareErrorFont`.

```

336 \fontfamily{cmr}
337 \fontseries{m}
338 \fontshape{n}
339 \fontsize{10}{10}

```

We now load the customizable parts of NFSS.

```

340 \InputIfFileExists{fonttext.cfg}
341     {\typeout{=====^^J%
342             ^^J%
343             Local config file fonttext.cfg used^^J%
344             ^^J%
345             =====}%
346     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
347     }
348     {\input{fonttext.ltx}}
349 \let\@addtofilelist\@gobble

```

Ditto for math although I don't think that we will get a lot of customisation :-)

```

350 \InputIfFileExists{fontmath.cfg}
351     {\typeout{=====^^J%
352             ^^J%

```

```

353             Local config file fontmath.cfg used^^J%
354             ^^J%
355             =====}%
356         \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
357     }
358     {\input{fontmath.ltx}}
359 \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`.

```

360 \InputIfFileExists{preload.cfg}
361     {\typeout{=====^^J%
362             ^^J%
363             Local config file preload.cfg used^^J%
364             ^^J%
365             =====}%
366         \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
367     }
368     {\input{preload.ltx}}
369 \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 370 \let\@acci\’ \let\@accii\’ \let\@acciii\=

\cal Here were the two old *⟨alphabet identifiers⟩*.

\mit
371 $\langle 2\text{kernel} \rangle$

File u

fontdef.dtx

j-latexreleasej [2019/12/17 v3.0e LaTeX Kernel (j-latexreleasej font setup)]

42 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.

43 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 \LaTeX 2 ϵ 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 \LaTeX versions. Thus, before sending in a bug report please try your test file with a \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 \LaTeX installations behave in the same way.

T1	Cork T _E X text encoding
OT1	old T _E X text encoding
U	unknown encoding
OML	old T _E X math letters encoding
OMS	old T _E X math symbols encoding
OMX	old T _E X math extension symbols encoding
TU	Unicode

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_EX 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_EX 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^AT_EX 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.

44 The docstrip modules

The following modules are used to direct **docstrip** in generating external files:

driver	produce a documentation driver file
text	produce the file fonttext.ltx
math	produce the file fontmath.ltx
cfgtext	produce a dummy fonttext.cfg file
cfgmath	produce a dummy fontmath.cfg file

A typical **docstrip** command file would then have entries like:

```
\generateFile{fonttext.ltx}{t}{\from{fontdef.dtx}{text}}
```

45 A driver for this document

The next bit of code contains the documentation driver file for T_EX, 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 \</driver>

```

46 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}

```

46.1 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}
14 \input{ts1enc.def}

```

```

15 \ifx\Umathchar\@undefined

```

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.

```

16 \fontencoding{OT1}
17 \def\rmsubstdefault{cmr}
18 \def\sfsubstdefault{cmss}
19 \def\ttsubstdefault{cmtt}
20 \LoadFontDefinitionFile{TS1}{cmr}
21 \else

```

Unicode.

```

22 \input {tuenc.def}
23 \fontencoding{TU}
24 \DeclareFontSubstitution{TU}{lmr}{m}{n}
25 \LoadFontDefinitionFile{TU}{lmr}
26 \LoadFontDefinitionFile{TU}{lmss}
27 \LoadFontDefinitionFile{TU}{lm tt}

```

```

28 \def\rmsubstdefault{lmr}
29 \def\sfsbstdefault{lmss}
30 \def\ttsbstdefault{lmtt}
31 \LoadFontDefinitionFile{TS1}{lmr}

32 \DeclareFontSubstitution{TU}{lmr}{m}{n}

```

End of Unicode branch.

```
33 \fi
```

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!

```
34 \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.

```

35 \DeclareFontSubstitution{T1}{cmr}{m}{n}
36 \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 $\text{\LaTeX 2}_{\epsilon}$ 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.

```

37 \begingroup
38 \nfss@catcodes
39 \input {t1cmr.fd}
40 \input {ot1cmr.fd}
41 \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 \TeX installation, while the amount of main memory is not a limiting factor at most installations.)

```

42 \begingroup
43 \nfss@catcodes
44 \input {ot1cmss.fd}
45 \input {ot1cmtt.fd}
46 \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.

```
47 \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.

46.2 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.

```

\encodingdefault The following three definitions set up the meaning for \rmfamily, \sffamily, and
\rmdefault       \ttfamily.
\sfdefault       48 \ifx\Umathchar\@undefined
\tddefault       49 \newcommand\encodingdefault{OT1}
                  50 \newcommand\rmdefault{cmr}
                  51 \newcommand\sfddefault{cmss}
                  52 \newcommand\tddefault{cmtt}
                  53 \else
                  54 \newcommand\encodingdefault{TU}
                  55 \newcommand\rmdefault{lmr}
                  56 \fontfamily{\rmdefault}
                  57 \newcommand\sfddefault{lmss}
                  58 \newcommand\tddefault{lm tt}
                  59 \fi
                  60 \</text>
                  61 \<latexrelease>\IncludeInRelease{2017/01/01}%
                  62 \<latexrelease>           {\encodingdefault}{TU encoding default}%
                  63 \<latexrelease>\ifx\Umathchar\@undefined
                  64 \<latexrelease>\renewcommand\encodingdefault{OT1}
                  65 \<latexrelease>\fontencoding{\encodingdefault}
                  66 \<latexrelease>\renewcommand\rmdefault{cmr}
                  67 \<latexrelease>\fontfamily{\rmdefault}
                  68 \<latexrelease>\renewcommand\sfddefault{cmss}
                  69 \<latexrelease>\renewcommand\tddefault{cmtt}
                  70 \<latexrelease>\else
                  71 \<latexrelease>\renewcommand\encodingdefault{TU}
                  72 \<latexrelease>%done in everyjob\fontencoding{\encodingdefault}
                  73 \<latexrelease>\renewcommand\rmdefault{lmr}
                  74 \<latexrelease>\fontfamily{\rmdefault}

```

```

75 \latexrelease\renewcommand\sfdefault{lmss}
76 \latexrelease\renewcommand\ttdefault{lmmt}
77 \latexrelease\fi
78 \latexrelease\EndIncludeInRelease
79 \latexrelease\IncludeInRelease{0000/00/00}%
80 \latexrelease\encodingdefault{TU encoding default}%
81 \latexrelease\fontencoding{OT1}
82 \latexrelease\renewcommand\encodingdefault{OT1}
83 \latexrelease\fontencoding{\encodingdefault}
84 \latexrelease\renewcommand\rmdefault{cmr}
85 \latexrelease\fontfamily{\rmdefault}
86 \latexrelease\renewcommand\sfdefault{cms}
87 \latexrelease\renewcommand\ttdefault{cmtt}
88 \latexrelease\EndIncludeInRelease
89 (*text)

\bfdefault Series changing commands are influenced by the following hooks.
\mddefault 90 \newcommand\bfdefault{b} % overwritten below
          91 \newcommand\mddefault{m}

\itdefault Shape changing commands use the following hooks.
\sldefault 92 \newcommand\itdefault{it}
\scdefault 93 \newcommand\sldefault{sl}
\updefault 94 \newcommand\scdefault{sc}
          95 \newcommand\updefault{up} % overwritten below

          96 \text
          97 (*text | latexrelease)
          98 \latexrelease\IncludeInRelease{2020/02/02}%
          99 \latexrelease\updefault{font defaults change}%
100 \renewcommand\updefault{up}
101 \renewcommand\bfdefault{b}
102 \text | latexrelease
103 \latexrelease\EndIncludeInRelease
104 \latexrelease\IncludeInRelease{0000/00/00}%
105 \latexrelease\updefault{font defaults change}%
106 \latexrelease
107 \latexrelease\renewcommand\updefault{n}
108 \latexrelease\renewcommand\bfdefault{bx}
109 \latexrelease\EndIncludeInRelease
110 (*text)

\familydefault Finally we have the hooks that describe the behaviour of the \normalfont com-
\seriesdefault mand. To stay portable, the definition of \encodingdefault should not be
\shapedefault changed and should match the setting above for \fontencoding. All other values
               can be set according to your taste.
111 \newcommand\familydefault{\rmdefault}
112 \newcommand\seriesdefault{\mddefault}

In previous releases \shapedefault pointed to \updefault which resolved to n,
but these days that is no longer the case (and up is wrong when you want to do
a reset. So we now use n explicitly.
113 % \changes{v3.0e}{2019/12/17}{Set \cs{shapedefault} explicitly to ‘n’}
114 \newcommand\shapedefault{n}

```

This finishes the low-level setup in `fonttext.ltx`.
115 `\text`

47 The fontmath.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

116 `(*math)`
117 `\typeout{=== Don't modify this file, use a .cfg file instead ===^J}`

47.1 The font encodings used

118 `\DeclareFontEncoding{OML}{-}{-}`
119 `\DeclareFontEncoding{OMS}{-}{-}`
120 `\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.

121 `\DeclareFontEncoding{U}{-}{\noaccents@}`

The encodings for math are next:

122 `\DeclareFontSubstitution{OML}{cmm}{m}{it}`
123 `\DeclareFontSubstitution{OMS}{cmsy}{m}{n}`
124 `\DeclareFontSubstitution{OMX}{cmex}{m}{n}`
125 `\DeclareFontSubstitution{U}{cmr}{m}{n}`

126 `\begingroup`
127 `\nfss@catcodes`
128 `\input {omlcmm.fd}`
129 `\input {omscmsy.fd}`
130 `\input {omxcmex.fd}`
131 `\input {ucmr.fd}`
132 `\endgroup`

47.1.1 Symbolfont and Alphabet declarations

We now define the basic symbol fonts used by \LaTeX . 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.

133 `\DeclareSymbolFont{operators} {OT1}{cmr} {m}{n}`
134 `\DeclareSymbolFont{letters} {OML}{cmm} {m}{it}`
135 `\DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}`
136 `\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}`

137 `\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}`
138 `\SetSymbolFont{letters} {bold}{OML}{cmm} {b}{it}`
139 `\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.

```

140 \DeclareSymbolFontAlphabet{\mathrm}      {operators}
141 \DeclareSymbolFontAlphabet{\mathnormal}{letters}
142 \DeclareSymbolFontAlphabet{\mathcal}      {symbols}
143 \DeclareMathAlphabet          {\mathbf}    {OT1}{cmr}{bx}{n}
144 \DeclareMathAlphabet          {\mathsf}    {OT1}{cmss}{m}{n}
145 \DeclareMathAlphabet          {\mathit}    {OT1}{cmr}{m}{it}
146 \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.

```

147 \SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
148 \SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}

```

47.2 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).

```

149 \DeclareMathSizes{5}{5}{5}{5}
150 \DeclareMathSizes{6}{6}{5}{5}
151 \DeclareMathSizes{7}{7}{5}{5}
152 \DeclareMathSizes{8}{8}{6}{5}
153 \DeclareMathSizes{9}{9}{6}{5}
154 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
155 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
156 \DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
157 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
158 \DeclareMathSizes{\@xvipt}{\@xvipt}{\@xiipt}{\@xpt}
159 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
160 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xvipt}

```

47.3 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 `InitTeX`. However, we repeat them here to have a complete setup which can be exchanged with another if desired.

47.3.1 The letters

```

161 \DeclareMathSymbol{a}{\mathalpha}{letters}{‘a}
162 \DeclareMathSymbol{b}{\mathalpha}{letters}{‘b}
163 \DeclareMathSymbol{c}{\mathalpha}{letters}{‘c}
164 \DeclareMathSymbol{d}{\mathalpha}{letters}{‘d}

```

```

165 \DeclareMathSymbol{e}{\mathalpha}{letters}{'e}
166 \DeclareMathSymbol{f}{\mathalpha}{letters}{'f}
167 \DeclareMathSymbol{g}{\mathalpha}{letters}{'g}
168 \DeclareMathSymbol{h}{\mathalpha}{letters}{'h}
169 \DeclareMathSymbol{i}{\mathalpha}{letters}{'i}
170 \DeclareMathSymbol{j}{\mathalpha}{letters}{'j}
171 \DeclareMathSymbol{k}{\mathalpha}{letters}{'k}
172 \DeclareMathSymbol{l}{\mathalpha}{letters}{'l}
173 \DeclareMathSymbol{m}{\mathalpha}{letters}{'m}
174 \DeclareMathSymbol{n}{\mathalpha}{letters}{'n}
175 \DeclareMathSymbol{o}{\mathalpha}{letters}{'o}
176 \DeclareMathSymbol{p}{\mathalpha}{letters}{'p}
177 \DeclareMathSymbol{q}{\mathalpha}{letters}{'q}
178 \DeclareMathSymbol{r}{\mathalpha}{letters}{'r}
179 \DeclareMathSymbol{s}{\mathalpha}{letters}{'s}
180 \DeclareMathSymbol{t}{\mathalpha}{letters}{'t}
181 \DeclareMathSymbol{u}{\mathalpha}{letters}{'u}
182 \DeclareMathSymbol{v}{\mathalpha}{letters}{'v}
183 \DeclareMathSymbol{w}{\mathalpha}{letters}{'w}
184 \DeclareMathSymbol{x}{\mathalpha}{letters}{'x}
185 \DeclareMathSymbol{y}{\mathalpha}{letters}{'y}
186 \DeclareMathSymbol{z}{\mathalpha}{letters}{'z}

187 \DeclareMathSymbol{A}{\mathalpha}{letters}{'A}
188 \DeclareMathSymbol{B}{\mathalpha}{letters}{'B}
189 \DeclareMathSymbol{C}{\mathalpha}{letters}{'C}
190 \DeclareMathSymbol{D}{\mathalpha}{letters}{'D}
191 \DeclareMathSymbol{E}{\mathalpha}{letters}{'E}
192 \DeclareMathSymbol{F}{\mathalpha}{letters}{'F}
193 \DeclareMathSymbol{G}{\mathalpha}{letters}{'G}
194 \DeclareMathSymbol{H}{\mathalpha}{letters}{'H}
195 \DeclareMathSymbol{I}{\mathalpha}{letters}{'I}
196 \DeclareMathSymbol{J}{\mathalpha}{letters}{'J}
197 \DeclareMathSymbol{K}{\mathalpha}{letters}{'K}
198 \DeclareMathSymbol{L}{\mathalpha}{letters}{'L}
199 \DeclareMathSymbol{M}{\mathalpha}{letters}{'M}
200 \DeclareMathSymbol{N}{\mathalpha}{letters}{'N}
201 \DeclareMathSymbol{O}{\mathalpha}{letters}{'O}
202 \DeclareMathSymbol{P}{\mathalpha}{letters}{'P}
203 \DeclareMathSymbol{Q}{\mathalpha}{letters}{'Q}
204 \DeclareMathSymbol{R}{\mathalpha}{letters}{'R}
205 \DeclareMathSymbol{S}{\mathalpha}{letters}{'S}
206 \DeclareMathSymbol{T}{\mathalpha}{letters}{'T}
207 \DeclareMathSymbol{U}{\mathalpha}{letters}{'U}
208 \DeclareMathSymbol{V}{\mathalpha}{letters}{'V}
209 \DeclareMathSymbol{W}{\mathalpha}{letters}{'W}
210 \DeclareMathSymbol{X}{\mathalpha}{letters}{'X}
211 \DeclareMathSymbol{Y}{\mathalpha}{letters}{'Y}
212 \DeclareMathSymbol{Z}{\mathalpha}{letters}{'Z}

```

47.3.2 The digits

```

213 \DeclareMathSymbol{0}{\mathalpha}{operators}{'0}
214 \DeclareMathSymbol{1}{\mathalpha}{operators}{'1}
215 \DeclareMathSymbol{2}{\mathalpha}{operators}{'2}
216 \DeclareMathSymbol{3}{\mathalpha}{operators}{'3}

```

```

217 \DeclareMathSymbol{4}{\mathalpha}{operators}{'4}
218 \DeclareMathSymbol{5}{\mathalpha}{operators}{'5}
219 \DeclareMathSymbol{6}{\mathalpha}{operators}{'6}
220 \DeclareMathSymbol{7}{\mathalpha}{operators}{'7}
221 \DeclareMathSymbol{8}{\mathalpha}{operators}{'8}
222 \DeclareMathSymbol{9}{\mathalpha}{operators}{'9}

```

47.3.3 Punctuation, brace, etc. keys

```

223 \DeclareMathSymbol{!}{\mathclose}{operators}{"21}
224 \DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
225 \DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
226 \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
227 \DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
228 \DeclareMathSymbol{.}{\mathord}{letters}{"3A}
229 \DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
230 \DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
231 \DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
232 \DeclareMathSymbol{?}{\mathclose}{operators}{"3F}

```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```

233 %\DeclareMathSymbol{(){\mathopen}{operators}{"28}
234 %\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
235 %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
236 %\DeclareMathSymbol{[]{\mathopen}{operators}{"5B}
237 %\DeclareMathSymbol{[]{\mathclose}{operators}{"5D}
238 %\DeclareMathSymbol{[]{\mathord}{symbols}{"6A}
239 %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
240 %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}

```

Should all of the following being activated by default? Probably not.

```

241 %\DeclareMathSymbol{'\}{\mathopen}{symbols}{"66}
242 %\DeclareMathSymbol{'\}{\mathclose}{symbols}{"67}
243 %\DeclareMathSymbol{'\}{\mathord}{symbols}{"6E} % \backslash
244 \mathcode'\ =8000 % \space
245 \mathcode'\ ' =8000 % ^\prime
246 \mathcode'\_ =8000 % \_

```

47.3.4 Delimitercodes for characters

[to be completed]

Finally, `IniTeX` sets all `\delcode` values to -1, except `\delcode' = 0`

```

247 \DeclareMathDelimiter{(){\mathopen}{operators}{"28}{largesymbols}{"00}
248 \DeclareMathDelimiter{)}{\mathclose}{operators}{"29}{largesymbols}{"01}
249 \DeclareMathDelimiter{[]{\mathopen}{operators}{"5B}{largesymbols}{"02}
250 \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.

```

251 \DeclareMathDelimiter{<}{\mathopen}{symbols}{"68}{largesymbols}{"0A}
252 \DeclareMathDelimiter{>}{\mathclose}{symbols}{"69}{largesymbols}{"0B}
253 \DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
254 \DeclareMathSymbol{>}{\mathrel}{letters}{"3E}

```


And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```

255 \DeclareMathDelimiter{/}{\mathord}{operators}{2F}{largesymbols}{OE}
256 \DeclareMathSymbol{/}{\mathord}{letters}{3D}

257 \DeclareMathDelimiter{|}{\mathord}{symbols}{6A}{largesymbols}{OC}

258 \expandafter\DeclareMathDelimiter\@backslashchar
259           {\mathord}{symbols}{6E}{largesymbols}{OF}

```

N.B. { and } should NOT get delcodes; otherwise parameter grouping fails!

47.4 Symbols accessed via control sequences

47.4.1 Greek letters

```

260 \DeclareMathSymbol{\alpha}{\mathord}{letters}{0B}
261 \DeclareMathSymbol{\beta}{\mathord}{letters}{0C}
262 \DeclareMathSymbol{\gamma}{\mathord}{letters}{0D}
263 \DeclareMathSymbol{\delta}{\mathord}{letters}{0E}
264 \DeclareMathSymbol{\epsilon}{\mathord}{letters}{0F}
265 \DeclareMathSymbol{\zeta}{\mathord}{letters}{10}
266 \DeclareMathSymbol{\eta}{\mathord}{letters}{11}
267 \DeclareMathSymbol{\theta}{\mathord}{letters}{12}
268 \DeclareMathSymbol{\iota}{\mathord}{letters}{13}
269 \DeclareMathSymbol{\kappa}{\mathord}{letters}{14}
270 \DeclareMathSymbol{\lambda}{\mathord}{letters}{15}
271 \DeclareMathSymbol{\mu}{\mathord}{letters}{16}
272 \DeclareMathSymbol{\nu}{\mathord}{letters}{17}
273 \DeclareMathSymbol{\xi}{\mathord}{letters}{18}
274 \DeclareMathSymbol{\pi}{\mathord}{letters}{19}
275 \DeclareMathSymbol{\rho}{\mathord}{letters}{1A}
276 \DeclareMathSymbol{\sigma}{\mathord}{letters}{1B}
277 \DeclareMathSymbol{\tau}{\mathord}{letters}{1C}
278 \DeclareMathSymbol{\upsilon}{\mathord}{letters}{1D}
279 \DeclareMathSymbol{\phi}{\mathord}{letters}{1E}
280 \DeclareMathSymbol{\chi}{\mathord}{letters}{1F}
281 \DeclareMathSymbol{\psi}{\mathord}{letters}{20}
282 \DeclareMathSymbol{\omega}{\mathord}{letters}{21}
283 \DeclareMathSymbol{\varepsilon}{\mathord}{letters}{22}
284 \DeclareMathSymbol{\vartheta}{\mathord}{letters}{23}
285 \DeclareMathSymbol{\varpi}{\mathord}{letters}{24}
286 \DeclareMathSymbol{\varrho}{\mathord}{letters}{25}
287 \DeclareMathSymbol{\varsigma}{\mathord}{letters}{26}
288 \DeclareMathSymbol{\varphi}{\mathord}{letters}{27}
289 \DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{00}
290 \DeclareMathSymbol{\Delta}{\mathalpha}{operators}{01}
291 \DeclareMathSymbol{\Theta}{\mathalpha}{operators}{02}
292 \DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{03}
293 \DeclareMathSymbol{\Xi}{\mathalpha}{operators}{04}
294 \DeclareMathSymbol{\Pi}{\mathalpha}{operators}{05}
295 \DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{06}
296 \DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{07}
297 \DeclareMathSymbol{\Phi}{\mathalpha}{operators}{08}
298 \DeclareMathSymbol{\Psi}{\mathalpha}{operators}{09}
299 \DeclareMathSymbol{\Omega}{\mathalpha}{operators}{0A}

```

47.4.2 Ordinary symbols

```

300 \DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
301 \DeclareMathSymbol{\imath}{\mathord}{letters}{"7B}
302 \DeclareMathSymbol{\jmath}{\mathord}{letters}{"7C}
303 \DeclareMathSymbol{\ell}{\mathord}{letters}{"60}
304 \DeclareMathSymbol{\wp}{\mathord}{letters}{"7D}
305 \DeclareMathSymbol{\Re}{\mathord}{symbols}{"3C}
306 \DeclareMathSymbol{\Im}{\mathord}{symbols}{"3D}
307 \DeclareMathSymbol{\partial}{\mathord}{letters}{"40}
308 \DeclareMathSymbol{\infty}{\mathord}{symbols}{"31}
309 \DeclareMathSymbol{\prime}{\mathord}{symbols}{"30}
310 \DeclareMathSymbol{\emptyset}{\mathord}{symbols}{"3B}
311 \DeclareMathSymbol{\nabla}{\mathord}{symbols}{"72}
312 \DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
313 \DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
314 \DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
315 \DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
316 \DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
317 \DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}

Alias:
318 % \let\not=\neg
319 \DeclareMathSymbol{\lnot}{\mathord}{symbols}{"3A}

320 \DeclareMathSymbol{\flat}{\mathord}{letters}{"5B}
321 \DeclareMathSymbol{\natural}{\mathord}{letters}{"5C}
322 \DeclareMathSymbol{\sharp}{\mathord}{letters}{"5D}
323 \DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{"7C}
324 \DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{"7D}
325 \DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{"7E}
326 \DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{"7F}

327 \DeclareRobustCommand\hbar{{\mathchar'26\mkern-9mu h}}
328 \DeclareRobustCommand\surd{{\mathchar"1270}}
329 \DeclareRobustCommand\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crrc
330 \not\mathrel{\mkern14mu}\crrc
331 \noalign{\nointerlineskip}
332 \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crrc}}}}

```

47.4.3 Large Operators

```

333 \DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{"60}
334 \DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{"57}
335 \DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{"56}
336 \DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{"55}
337 \DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{"54}
338 \DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{"53}
339 \DeclareMathSymbol{\intop}{\mathop}{largesymbols}{"52}
340 \DeclareRobustCommand\int{\intop\nolimits}
341 \DeclareMathSymbol{\prod}{\mathop}{largesymbols}{"51}
342 \DeclareMathSymbol{\sum}{\mathop}{largesymbols}{"50}
343 \DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{"4E}
344 \DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{"4C}
345 \DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{"4A}
346 \DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{"48}
347 \DeclareRobustCommand\oint{\ointop\nolimits}

```

```

348 \DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{46}
349 \DeclareMathSymbol{\smallint}{\mathop}{symbols}{73}

```

47.4.4 Binary symbols

```

350 \DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{2F}
351 \DeclareMathSymbol{\triangleright}{\mathbin}{letters}{2E}
352 \DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{34}
353 \DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{35}

```

Alias:

```

354 % \let \varbigtriangledown \bigtriangledown
355 % \let \varbigtriangleup \bigtriangleup
356 \DeclareMathSymbol{\varbigtriangleup}{\mathbin}{symbols}{34}
357 \DeclareMathSymbol{\varbigtriangledown}{\mathbin}{symbols}{35}

```

These last two synonyms are needed because the `stmaryrd` package redefines them as Operators.

```

358 \DeclareMathSymbol{\wedge}{\mathbin}{symbols}{5E}
359 \DeclareMathSymbol{\vee}{\mathbin}{symbols}{5F}

```

Alias:

```

360 % \let \land = \wedge
361 % \let \lor = \vee
362 \DeclareMathSymbol{\land}{\mathbin}{symbols}{5E}
363 \DeclareMathSymbol{\lor}{\mathbin}{symbols}{5F}
364 \DeclareMathSymbol{\cap}{\mathbin}{symbols}{5C}
365 \DeclareMathSymbol{\cup}{\mathbin}{symbols}{5B}
366 \DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{7A}
367 \DeclareMathSymbol{\dagger}{\mathbin}{symbols}{79}
368 \DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{75}
369 \DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{74}
370 \DeclareMathSymbol{\uplus}{\mathbin}{symbols}{5D}
371 \DeclareMathSymbol{\amalg}{\mathbin}{symbols}{71}
372 \DeclareMathSymbol{\diamond}{\mathbin}{symbols}{05}
373 \DeclareMathSymbol{\bullet}{\mathbin}{symbols}{0F}
374 \DeclareMathSymbol{\wr}{\mathbin}{symbols}{6F}
375 \DeclareMathSymbol{\div}{\mathbin}{symbols}{04}
376 \DeclareMathSymbol{\odot}{\mathbin}{symbols}{0C}
377 \DeclareMathSymbol{\oslash}{\mathbin}{symbols}{0B}
378 \DeclareMathSymbol{\otimes}{\mathbin}{symbols}{0A}
379 \DeclareMathSymbol{\ominus}{\mathbin}{symbols}{09}
380 \DeclareMathSymbol{\oplus}{\mathbin}{symbols}{08}
381 \DeclareMathSymbol{\mp}{\mathbin}{symbols}{07}
382 \DeclareMathSymbol{\pm}{\mathbin}{symbols}{06}
383 \DeclareMathSymbol{\circ}{\mathbin}{symbols}{0E}
384 \DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{0D}
385 \DeclareMathSymbol{\setminus}{\mathbin}{symbols}{6E}
386 \DeclareMathSymbol{\cdot}{\mathbin}{symbols}{01}
387 \DeclareMathSymbol{\ast}{\mathbin}{symbols}{03}
388 \DeclareMathSymbol{\times}{\mathbin}{symbols}{02}
389 \DeclareMathSymbol{\star}{\mathbin}{letters}{3F}

```

47.4.5 Relations

```

390 \DeclareMathSymbol{\propto}{\mathrel}{symbols}{2F}
391 \DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{76}

```

```

392 \DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{"77}
393 \DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
394 \DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
395 \DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
396 \DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
397 \DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
398 \DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
399 \DeclareMathSymbol{\nwarrow}{\mathrel}{symbols}{"2D}
400 \DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
401 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
402 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
403 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
404 \DeclareRobustCommand\neq{\not=}

```

As `\neq` is robust we should not use `\let` to define `\ne` as then then it would change if `\neq` changes.

```
405 \DeclareRobustCommand\ne{\not=}
```

It would ok to use `\let` for those declared by `\DeclareMathSymbol` but for a cleaner interface we avoid it always (just in case the internals change).

```

406 \DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
407 \DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}

```

Alias:

```

408 % \let\le=\leq
409 % \let\ge=\geq
410 \DeclareMathSymbol{\le}{\mathrel}{symbols}{"14}
411 \DeclareMathSymbol{\ge}{\mathrel}{symbols}{"15}
412 \DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
413 \DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
414 \DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
415 \DeclareMathSymbol{\succeq}{\mathrel}{symbols}{"17}
416 \DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
417 \DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
418 \DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
419 \DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
420 \DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
421 \DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
422 \DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}

```

Alias:

```

423 % \let\owns=\ni
424 \DeclareMathSymbol{\owns}{\mathrel}{symbols}{"33}
425 \DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
426 \DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
427 \DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
428 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"24}
429 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"20}
430 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"21}

```

Alias:

```

431 % \let\gets=\leftarrow
432 % \let\to=\rightarrow
433 \DeclareMathSymbol{\gets}{\mathrel}{symbols}{"20}
434 \DeclareMathSymbol{\to}{\mathrel}{symbols}{"21}
435 \DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}

```

```

436 \DeclareRobustCommand\mapsto{\mapstochar\rightarrow}
437 \DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
438 \DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
439 \DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
440 \DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
441 \DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
442 \DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}
443 \DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
444 \DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
445 \DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
446 \DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
447 \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.

```

448 \DeclareRobustCommand
449 \cong{\mathrel{\mathpalette@vereq\sim}} % congruence sign
450 \def@vereq#1#2{\lower.5p@\vbox{\lineskiplimit\maxdimen\lineskip-.5p@
451 \ialign{$\m@th#1\hfil##\hfil$\crrcr#2\crrcr\crrcr}}}
452 \DeclareRobustCommand
453 \notin{\mathrel{\m@th\mathpalette@c@ncel\in}}
454 \def@c@ncel#1#2{\m@th\oalign{$\hfil#1\mkern1mu\hfil$\crrcr$#1#2$}}
455 \DeclareRobustCommand
456 \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
457 \def\rlh@#1{\vcenter{\m@th\hbox{\oalign{\raise2pt
458 \hbox{$#1\rightharpoonup$}\crrcr
459 $#1\leftharpoondown$}}}}
460 \DeclareRobustCommand
461 \doteq{\buildrel\textstyle.\over=}

```

47.4.6 Arrows

```

462 \DeclareRobustCommand
463 \joinrel{\mathrel{\mkern-3mu}}
464 \DeclareRobustCommand
465 \relbar{\mathrel{\smash-}} % \smash, because -
466 % 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).

```

467 \DeclareRobustCommand
468 \Relbar{\mathrel{=}}
469 \DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
470 \DeclareRobustCommand\hookrightarrow{\lhook\joinrel\rightarrow}
471 \DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
472 \DeclareRobustCommand\hookleftarrow{\leftarrow\joinrel\rhook}
473 \DeclareRobustCommand

```

```

474 \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
475 \DeclareRobustCommand
476 \models{\mathrel{||}\joinrel\Relbar}
477 \DeclareRobustCommand
478 \Longrightarrow{\Relbar\joinrel\Rightarrow}

```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make then robust.

```

479 \DeclareRobustCommand\longrightarrow
480 {\relbar\joinrel\rightarrow}
481 \DeclareRobustCommand\longleftarrow
482 {\leftarrow\joinrel\relbar}
483 \DeclareRobustCommand
484 \Longleftarrow{\Leftarrow\joinrel\Relbar}
485 \DeclareRobustCommand
486 \longmapsto{\mapstochar\longrightarrow}
487 \DeclareRobustCommand
488 \longlefttrightarrow{\leftarrow\joinrel\rightarrow}
489 \DeclareRobustCommand
490 \Longlefttrightarrow{\Leftarrow\joinrel\Rightarrow}
491 \DeclareRobustCommand
492 \iff{\;\Longlefttrightarrow\;}

```

47.4.7 Punctuation symbols

```

493 \DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
494 \DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
495 \DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}

```

This is commented out, since `\ldots` is now defined in `ltoutenc.dtx`.

```

496 %\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
497 %\DeclareRobustCommand\ldots
498 % {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots$,}$\fi}
499 \DeclareRobustCommand
500 \cdots{\mathinner{\cdotp\cdotp\cdotp}}
501 \DeclareRobustCommand
502 \vdots{\vbox{\baselineskip4p@ \lineskiplimit\z@
503 \kern6p@\hbox{.}\hbox{.}\hbox{.}}}
504 \DeclareRobustCommand
505 \ddots{\mathinner{\mkern1mu\raise7p@
506 \vbox{\kern7p@\hbox{.}}\mkern2mu
507 \raise4p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}

```

47.4.8 Math accents

```

508 \DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
509 \DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}
510 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
511 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
512 \DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
513 \DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
514 \DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
515 \DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
516 \DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
517 \DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
518 \DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}

```

```
519 \DeclareMathAccent{\widehat}{\mathord}{\largesymbols}{62}
```

For some reason plain T_EX 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:

```
520 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{17}
```

47.4.9 Radicals

```
521 \DeclareMathRadical{\sqrtsign}{symbols}{70}{largesymbols}{70}
```

47.4.10 Over and under something, etc

```
522 \DeclareRobustCommand\overrightarrow[1]{\vbox{\m@th\ialign{##\crrc
523   \rightarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}
524   $\hfil\displaystyle{#1}\hfil$\crrc}}
525 \DeclareRobustCommand\overleftarrow[1]{\vbox{\m@th\ialign{##\crrc
526   \leftarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}%
527   $\hfil\displaystyle{#1}\hfil$\crrc}}
528 \DeclareRobustCommand\overbrace[1]{\mathop{\vbox{\m@th\ialign{##\crrc\noalign{\kern3\p@}%
529   \downbracefill\crrc\noalign{\kern3\p@\nointerlineskip}%
530   $\hfil\displaystyle{#1}\hfil$\crrc}}\limits}
531   $\hfil\displaystyle{#1}\hfil$\crrc}}\limits}
532 \DeclareRobustCommand\underbrace[1]{\mathop{\vtop{\m@th\ialign{##\crrc
533   $\hfil\displaystyle{#1}\hfil$\crrc
534   \noalign{\kern3\p@\nointerlineskip}%
535   \upbracefill\crrc\noalign{\kern3\p@}}}\limits}
```

(quite a waste of tokens, IMHO — Frank)

```
536 \DeclareRobustCommand\skew[3]
537   {\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
538   #2{\mkern-\muskip\z@#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
539 \DeclareRobustCommand\rightarrowfill{$\m@th\smash-\mkern-7mu%
540   \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
541   \mkern-7mu\mathord\rightarrow$}
542 \DeclareRobustCommand\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
543   \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
544   \mkern-7mu\smash-$}
545 \DeclareMathSymbol{\braceld}{\mathord}{\largesymbols}{7A}
546 \DeclareMathSymbol{\bracerd}{\mathord}{\largesymbols}{7B}
547 \DeclareMathSymbol{\bracelu}{\mathord}{\largesymbols}{7C}
548 \DeclareMathSymbol{\braceru}{\mathord}{\largesymbols}{7D}
549 \DeclareRobustCommand\downbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
550   \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru
551   \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd$}
552 \DeclareRobustCommand\upbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
553   \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
554   \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}
```

47.4.11 Delimiters

```
555 \DeclareMathDelimiter{\lmoustache} % top from (, bottom from )
556   {\mathopen}{\largesymbols}{7A}{\largesymbols}{40}
557 \DeclareMathDelimiter{\rmoustache} % top from ), bottom from (
558   {\mathclose}{\largesymbols}{7B}{\largesymbols}{41}
559 \DeclareMathDelimiter{\arrowvert} % arrow without arrowheads
560   {\mathord}{symbols}{6A}{\largesymbols}{3C}
561 \DeclareMathDelimiter{\Arrowvert} % double arrow without arrowheads
```

```

562 {\mathord}{symbols}{"6B}{largesymbols}{"3D}
563 \DeclareMathDelimiter{\Vert}
564 {\mathord}{symbols}{"6B}{largesymbols}{"0D}

\DeclareMathDelimiter produces a command that is robust (with an internal
macro containing the payload) so we should not use \let for making an alias
565 %\let\|= \Vert
566 \DeclareMathDelimiter{\|}
567 {\mathord}{symbols}{"6B}{largesymbols}{"0D}
568 \DeclareMathDelimiter{\vert}
569 {\mathord}{symbols}{"6A}{largesymbols}{"0C}
570 \DeclareMathDelimiter{\uparrow}
571 {\mathrel}{symbols}{"22}{largesymbols}{"78}
572 \DeclareMathDelimiter{\downarrow}
573 {\mathrel}{symbols}{"23}{largesymbols}{"79}
574 \DeclareMathDelimiter{\updownarrow}
575 {\mathrel}{symbols}{"6C}{largesymbols}{"3F}
576 \DeclareMathDelimiter{\Uparrow}
577 {\mathrel}{symbols}{"2A}{largesymbols}{"7E}
578 \DeclareMathDelimiter{\Downarrow}
579 {\mathrel}{symbols}{"2B}{largesymbols}{"7F}
580 \DeclareMathDelimiter{\Updownarrow}
581 {\mathrel}{symbols}{"6D}{largesymbols}{"77}
582 \DeclareMathDelimiter{\backslash} % for double coset G\backslash H
583 {\mathord}{symbols}{"6E}{largesymbols}{"0F}
584 \DeclareMathDelimiter{\rangle}
585 {\mathclose}{symbols}{"69}{largesymbols}{"0B}
586 \DeclareMathDelimiter{\langle}
587 {\mathopen}{symbols}{"68}{largesymbols}{"0A}
588 \DeclareMathDelimiter{\rbrace}
589 {\mathclose}{symbols}{"67}{largesymbols}{"09}
590 \DeclareMathDelimiter{\lbrace}
591 {\mathopen}{symbols}{"66}{largesymbols}{"08}
592 \DeclareMathDelimiter{\rceil}
593 {\mathclose}{symbols}{"65}{largesymbols}{"07}
594 \DeclareMathDelimiter{\lceil}
595 {\mathopen}{symbols}{"64}{largesymbols}{"06}
596 \DeclareMathDelimiter{\rfloor}
597 {\mathclose}{symbols}{"63}{largesymbols}{"05}
598 \DeclareMathDelimiter{\lfloor}
599 {\mathopen}{symbols}{"62}{largesymbols}{"04}

```

`\lgroup` There are three plain \TeX delimiters which are not fully supported by NFSS,
`\rgroup` since they partly point into a bold cmr font. Allocating a full symbol font, just
`\bracevert` to have three delimiters seems a bit too much given the limited space available.
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 `cmtt/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.)

```

600 \DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
601 {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
602 \DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
603 {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}

```



```

604 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
605      {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}

```

47.5 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 606 \DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
\mathdollar 607 \DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
\mathsterling 608 \DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
\mathunderscore 609 \DeclareRobustCommand\mathsterling{\mathit{\mathchar"7024}}
610 \DeclareRobustCommand\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}

\mathellipsis This is plain TEX's \ldots.
611 \DeclareRobustCommand\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%

```

47.6 Other special functions and parameters

47.6.1 Biggggg

```

612 \</math>
613 \<math>| latexrelease>
614 \<latexrelease> \<IncludeInRelease{2018/12/01}>%
615 \<latexrelease>      {\Big}{Start LR-mode}%
616 \DeclareRobustCommand\big[1]{\leavevmode@ifvmode
617   {\hbox{$\left#1\vbox to8.5\p@{\right.\n@space$}}}
618 \DeclareRobustCommand\Big[1]{\leavevmode@ifvmode
619   {\hbox{$\left#1\vbox to11.5\p@{\right.\n@space$}}}
620 \DeclareRobustCommand\bigg[1]{\leavevmode@ifvmode
621   {\hbox{$\left#1\vbox to14.5\p@{\right.\n@space$}}}
622 \DeclareRobustCommand\Bigg[1]{\leavevmode@ifvmode
623   {\hbox{$\left#1\vbox to17.5\p@{\right.\n@space$}}}
624 \</math>| latexrelease>
625 \<latexrelease> \<EndIncludeInRelease>
626 \<latexrelease> \<IncludeInRelease{0000/00/00}>%
627 \<latexrelease>      {\Big}{Start LR-mode}%
628 \<latexrelease> \<def>\big#1{{\hbox{$\left#1\vbox to8.5\p@{\right.\n@space$}}}
629 \<latexrelease> \<def>\Big#1{{\hbox{$\left#1\vbox to11.5\p@{\right.\n@space$}}}
630 \<latexrelease> \<def>\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{\right.\n@space$}}}
631 \<latexrelease> \<def>\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{\right.\n@space$}}}
632 \<latexrelease> \<EndIncludeInRelease>
633 \<math>
634 \<def>\n@space{\nulldelimiterspace\z@ \m@th}

```

47.6.2 The log-like functions

```

\operator@font The \operator@font determines the symbol font used for log-like functions.
635 \def\operator@font{\mathgroup\symoperators}

```

47.6.3 Parameters

```
636 \thinmuskip=3mu
637 \medmuskip=4mu plus 2mu minus 4mu
638 \thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in `fontmath.ltx`.

```
639 \end{math}
```

48 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.

```
640 (*cfgtext | cfgmath | cfgprel)
641 %%
642 %%
643 %%
644 %% Load the standard setup:
645 %%
646 \input{fonttext.ltx}
647 \input{fontmath.ltx}
648 \input{preload.ltx}
649 %%
650 %% Small changes could go here; see documentation in cfgguide.tex for
651 %% allowed modifications.
652 %%
653 %% In particular it is not allowed to misuse this configuration file
654 %% to modify internal LaTeX commands!
655 %%
656 %% If you use this file as the basis for configuration please change
657 %% the \ProvidesFile lines to clearly identify your modification, e.g.,
658 %%
659 \input{fonttext.ltx} \ProvidesFile{fonttext.cfg}[2001/06/01
660 \input{fontmath.ltx} \ProvidesFile{fontmath.cfg}[2001/06/01
661 \input{preload.ltx} \ProvidesFile{preload.cfg}[2001/06/01
662 %% Customised local font setup]
663 %%
664 %%
665 \end{cfgtext | cfgmath | cfgprel}
```

File v

preload.dtx

49 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by L^AT_EX 2_ε). 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
depreloa.xpt	preload of DC fonts for 10pt size
depreloa.xip	preload of DC fonts for 11pt size
depreloa.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.

50 Customization

You can customize the preloaded fonts in your L^AT_EX 2_ε 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^AT_EX 2_ε 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^AT_EX 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.

51 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.

52 A driver for this document

The next bit of code contains the documentation driver file for T_EX, 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}
```

53 The code

We begin by loading the math extension font (cmex10) and the L^AT_EX 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\tencirc=lcircle10 \font\tencircw=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 (+xipt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
30 (+xipt & 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 w

ltfntcmd.dtx

Abstract

The commands defined in this file `ltfntcmd` are part of the kernel code for L^AT_EX 2_ε/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.

54 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_EX system and for several reasons it is better to avoid them on the user level whenever possible. In L^AT_EX3 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`.¹¹ The default definition for this list is produced by the following.

```
\newcommand \nocorrlist {,.}
```

¹¹Any 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>\textnormal{..}</code>	<code>\normalfont</code>	Typeset argument in normal family
<code>\textrm{..}</code>	<code>\rmfamily</code>	Typeset argument in roman family
<code>\textsf{..}</code>	<code>\sffamily</code>	Typeset argument in sans serif family
<code>\texttt{..}</code>	<code>\ttfamily</code>	Typeset argument in typewriter family
<code>\textmd{..}</code>	<code>\mdseries</code>	Typeset argument in medium series
<code>\textbf{..}</code>	<code>\bfseries</code>	Typeset argument in bold 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^AT_EX 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^AT_EX'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 `\/`.

55 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@ic1 #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}

textulc
textsw 25 \if2kernel
textssc 26 \if*2kernel | latexrelease
27 \if latexrelease \IncludeInRelease{2020/02/02}%
28 \if latexrelease {\textulc}{Additional text commands}%
29 \DeclareTextFontCommand{\textulc}{\ulcshape}
30 \DeclareTextFontCommand{\textsw}{\swshape}
31 \DeclareTextFontCommand{\textssc}{\sscshape}
32 \if2kernel | latexrelease
33 \if latexrelease \EndIncludeInRelease
34 \if latexrelease \IncludeInRelease{0000/00/00}%
35 \if latexrelease {\textulc}{Additional text commands}%
36 \if latexrelease
37 \if latexrelease \let\textulc\@undefined
38 \if latexrelease \let\textsw\@undefined
39 \if latexrelease \let\textssc\@undefined
40 \if latexrelease \EndIncludeInRelease
41 \if*2kernel

\emph Finally we have the \em font change declaration of LATEX. The corresponding
definition with argument is
42 \DeclareTextFontCommand{\emph}{\em}

\nocorr This is just a label, so it does nothing; it should also be unexpandable.
43 \let \nocorr \relax

```

`\check@icl` We define these defaults in case some error causes them to be expanded at the wrong time.

```
44 \let \check@icl \@empty
45 \let \check@icr \@empty
```

`\text@command` This checks for a `\nocorr` as the first token in its argument and also for one in any other position not protected within braces (the latter is treated as if it were at the end of the argument).

`\check@nocorr@`

Is this the correct action in the ‘empty’ case? It is efficient but typographically it is, strictly, incorrect!

```
46 \def \text@command #1{%
47   \def \reserved@a {#1}%
48   \ifx \reserved@a \@empty
49     \let \check@icl \@empty
50     \let \check@icr \@empty
51   \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.

```
52 %   \def \reserved@b { }%
53 %   \ifx \reserved@a \reserved@b
54   \ifx \reserved@a \space
55     \let \check@icl \@empty
56     \let \check@icr \@empty
57   \else
58     \check@nocorr@ #1\nocorr\@nil
59   \fi
60 \fi
61 }
62 \def \check@nocorr@ #1#2\nocorr#3\@nil {%
```

The two checks are initialised here to their values in the normal case.

```
63 \let \check@icl \maybe@ic
64 \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi}%
65 \def \reserved@a {\nocorr}%
66 \def \reserved@b {#1}%
67 \def \reserved@c {#3}%
68 \ifx \reserved@a \reserved@b
69   \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.

```
70   \let \check@icl \@empty
71   \else
```

Otherwise there is a `\nocorr` both at the start and elsewhere, so no italic corrections should be added.

```
72   \let \check@icl \@empty
73   \let \check@icr \@empty
74   \fi
75 \else
76   \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.

```
77 \else
```

In this case there is no `\nocorr` at the start but there is one elsewhere, so no `\aftergroup` is needed.

```
78 \let \check@icr \@empty
79 \fi
80 \fi
81 }
```

`\ifmaybe@ic` Switch used solely within `\maybe@ic` not interfering with other switches.

```
82 \newif\ifmaybe@ic
```

`\maybe@ic` These macros implement the italic correction.

```
\maybe@ic@ 83 \def \maybe@ic {\futurelet \@let@token \maybe@ic@}
84 \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?

```
85 \ifdim \fontdimen \@ne \font > \z@
86 \else
87 \maybe@ictrue
```

It would be possible, but probably not worthwhile, to continue the forward scan beyond any closing braces.

```
88 \expandafter \@tfor \expandafter \reserved@a \expandafter = %
89 \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.

```
90 \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`.

```
91 \ifmaybe@ic \sw@slant \fi
92 \fi
93 }
```

`\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.

```
94 \def \t@st@ic {%
95 \expandafter \let \expandafter \reserved@b \expandafter = \reserved@a \relax
96 \ifx \reserved@b \@let@token
```

If they are the same we record the fact and jump out of the loop.

```

97     \maybe@icfalse
98     \@break@tfor
99     \fi
100 }

```

`\sw@slant` The definition of the mysterious `\sw@slant` command is as follows.
`\fix@penalty` 101 `\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.

```

102 \ifdim \lastskip=\z@
103   \fix@penalty
104 \else
105   \skip@ \lastskip
106   \unskip
107   \fix@penalty
108   \hskip \skip@
109 \fi
110 }

```

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^AT_EX code?

```

111 \def \fix@penalty {%
112   \ifnum \lastpenalty=\z@
113     \@@italiccorr
114   \else
115     \count@ \lastpenalty
116     \unpenalty
117     \@@italiccorr
118     \penalty \count@
119   \fi
120 }

```

`\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.

```

121 \def \nocorrlist {,}

```

`\nfss@text` This command will by default behave like a L^AT_EX `\mbox` but may be redefined by packages such as `amstext.sty` to be a bit cleverer.

```
122 \ifx \nfss@text\undefined
123   \def \nfss@text {\leavevmode\hbox}
124 \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}
```

```
125 \def \DeclareOldFontCommand #1#2#3{%
126   \DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
127 }
```

`\@fontswitch` These two commands actually do the necessary tests and declarative font- or alphabet-changing.

`\@@math@egroup`

`\@@math@egroup`

```
128 \def \@fontswitch #1#2{%
129   \ifmmode
130     \let \math@bgroup \relax
131     \def \math@egroup {\let \math@bgroup \@@math@bgroup
132                       \let \math@egroup \@@math@egroup}%
133   \else
134     \let \math@bgroup \relax
135     \def \math@egroup {\let \math@bgroup \@@math@bgroup
136                       \let \math@egroup \@@math@egroup}%
137 }
```

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).

```
133   #2\relax
134   \else
135     #1%
136   \fi
137 }
138 \let \@@math@bgroup \math@bgroup
139 \let \@@math@egroup \math@egroup
```

These commands are available only in the preamble.

```
140 \onlypreamble \DeclareTextFontCommand
141 \onlypreamble \DeclareOldFontCommand
```

56 Initialization

```
\normalsize This is defined to produce an error.
142 \def\normalsize{%
143   \@latex@error {The font size command \protect\normalsize\space
144                 is not defined:\MessageBreak
145                 there is probably something wrong with
146                 the class file}\@eha
147 }
148 </2ekernel>
```

File x

lttextcomp.dtx

This file contains the implementation for accessing the glyphs provided by the TS1 encoding (Text Companion Encoding). This is now offered as part of the kernel and so the `textcomp` package which used to provide the definitions is now mainly needed for compatibility reasons (and doesn't do much any more).

`\oldstylenums` Preserve the old definition of `\oldstylenums` under a different name.
`\legacyoldstylenums` 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.

```
1 (*2ekernel | latexrelease)
2 (latexrelease)\IncludeInRelease{2020/02/02}%
3 (latexrelease)          {\oldstylenums}{Old style numerals}%
4 \DeclareRobustCommand\legacyoldstylenums[1]{%
5   \begingroup
```

Provide spacing using the interword space of the current font.

```
6   \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^AT_EX fonts even though they only contain substitutions).

```
7   \usefont{OML}{\rmdefault}{\f@series}{it}%
8   \mathgroup\symletters #1%
9   \endgroup
10 }
```

And here is the improved one that adjusts depending on surroundings.

```
11 \DeclareRobustCommand\oldstylenums[1]{%
12   \begingroup
13   \ifmmode
14     \mathgroup\symletters #1%
15   \else
```

The `\CheckEncodingSubset` is discussed below.

```
16   \CheckEncodingSubset\@use@text@encoding{TS1}\tc@oldstylesubst2{\#1}%
17   \fi
18   \endgroup
19 }
```

The helper to select the substitution if needed.

```
20 \def\tc@oldstylesubst#1{%
21   \tc@errorwarn
22     {Oldstyle digits unavailable for
23     family \f@family.\MessageBreak
24     Default oldstyle digits used instead}\@eha
25   \bgroup
26     \expand@font@defaults
```


The substitution defaults are provided in the file `fonttext.ltx`.

```

27     \ifx\f@family\rmdef@ult
28         \fontfamily\rmsubstdefault
29     \else\ifx\f@family\sfddef@ult
30         \fontfamily\sfsbstdefault
31     \else\ifx\f@family\ttdef@ult
32         \fontfamily\ttsbstdefault
33     \else
34         \fontfamily\textcompsubstdefault
35     \fi\fi\fi
36     \fontencoding{TS1}\selectfont#1%
37 \egroup
38 }

```

`\textcompsubstdefault` Here is the default for the “unknown” case:

```

39 \def\textcompsubstdefault{\rmsubstdefault}

40 </2ekernel | latexrelease>
41 <latexrelease>\EndIncludeInRelease
42 <latexrelease>\IncludeInRelease{0000/00/00}%
43 <latexrelease>           {\oldstylenums}{Old style numerals}%
44 <latexrelease>
45 <latexrelease>\DeclareRobustCommand\oldstylenums[1]{%
46 <latexrelease>     \begingroup
47 <latexrelease>     \spaceskip\fontdimen\tw@\font
48 <latexrelease>     \usefont{OML}{\rmdefault}{\f@series}{it}%
49 <latexrelease>     \mathgroup\symbols #1%
50 <latexrelease>     \endgroup
51 <latexrelease>}
52 <latexrelease>\let\legacyoldstylenums\@undefined
53 <latexrelease>\def\textcompsubstdefault{cmr}
54 <latexrelease>
55 <latexrelease>\EndIncludeInRelease

```

Everything else in this file got introduced 2020/02/02, so we do a single rollback (for now).

```

56 <*2ekernel>
57 </2ekernel>
58 <*2ekernel | latexrelease>
59 <latexrelease>\IncludeInRelease{2020/02/02}%
60 <latexrelease>     {\DeclareEncodingSubset}{Text companion symbols}%

```

`\DeclareEncodingSubset` The declaration takes 3 mandatory arguments: an *encoding* for which a subsetting is wanted (currently always TS1, and most likely forever), the *font family* for which we declare the subset and finally the *subset* number (between 0 (all of the encoding is supported) and 9 many glyphs are missing).

For TS1 the numbers have been chosen in a way that most fonts can be fairly correctly categorized, but the default settings are always conservative, that is they may claim that less glyphs are supported than there actually are.

As these days many font families are set up to end in -LF (lining figures), -OsF (oldstyle figures), etc. the declaration supports a shortcut: if the *font family* name ends in -* then the star gets replaced by these common ending, e.g.,

```
\DeclareEncodingSubset{TS1}{Alegreya-*}{2}
```

is the same as writing

```
\DeclareEncodingSubeset{TS1}{Alegreya-LF}{2}
\DeclareEncodingSubeset{TS1}{Alegreya-OfF}{2}
\DeclareEncodingSubeset{TS1}{Alegreya-TLF}{2}
\DeclareEncodingSubeset{TS1}{Alegreya-TOfF}{2}
```

If only some are needed then one can define them individually but in many cases all four are wanted, hence the shortcut.

The coding of the declaration has no error checking as it is mostly for internal use.

```
61 \def\DeclareEncodingSubset#1#2{%
62   \DeclareEncodingSubset@aux{#1}#2*\DeclareEncodingSubset@aux
63 }
64 \def\DeclareEncodingSubset@aux#1#2*#3\DeclareEncodingSubset@aux#4{%
```

if #3 is empty then there was no star, otherwise we we define all four variants.

```
65   \expandafter\ifx\expandafter X\detokenize{#3}X%
66     \@DeclareEncodingSubset{#1}{#2}{#4}%
67   \else
68     \@DeclareEncodingSubset{#1}{#2LF}{#4}%
69     \@DeclareEncodingSubset{#1}{#2TLF}{#4}%
70     \@DeclareEncodingSubset{#1}{#2OfF}{#4}%
71     \@DeclareEncodingSubset{#1}{#2TOfF}{#4}%
72   \fi
73 }
```

The subset info is stored in a command with the name `\family:subset` so if that already exists we change otherwise declare a subset.

```
74 \def\@DeclareEncodingSubset#1#2#3{%
75   \ifundefined{#1:#2}%
76     {\@font@info{Setting #2 sub-encoding to #1/#3}}%
77     {\@font@info{Changing #2 sub-encoding to #1/#3}}%
78   \@namedef{#1:#2}{#3}}
```

Any reason to allow those in the middle of documents?

```
79 \@onlypreamble\DeclareEncodingSubset
80 \@onlypreamble\DeclareEncodingSubset@aux
81 \@onlypreamble\@DeclareEncodingSubset
```

`\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.

```

82 \def\CheckEncodingSubset#1#2#3#4#5{%
83   \ifnum #4>%
84     \expandafter\ifx\csname #2:\f@family\endcsname\relax
85       0\csname #2:\endcsname
86     \else
87       \csname #2:\f@family\endcsname
88     \fi
89   \relax
90   \expandafter\@firstoftwo
91 \else
92   \expandafter\@secondoftwo
93 \fi
94   {#1{#2}}{#3}%
95   #5%
96 }

```

To set up the glyphs for the subsets we need a number helpers.

`\tc@errorwarn` To we produce errors, warnings, or only info in the transcripts if glyphs require substitutions? By default it is “info” only. With the `textcomp` package that can be changed.

```

97 \def\tc@errorwarn#1#2{\@latex@info{#1}}

```

`\tc@subst`

```

98 \def\tc@subst#1{%
99   \tc@errorwarn
100   {Symbol \string#1 not provided by\MessageBreak
101     font family \f@family\space
102     in TS1 encoding.\MessageBreak Default family used instead}\@eha
103   \bgroup
104     \expand@font@defaults
105     \ifx\f@family\rmdef@ult
106       \fontfamily\rmsubstdefault
107     \else\ifx\f@family\sfsdef@ult
108       \fontfamily\sfsbstdefault
109     \else\ifx\f@family\ttdef@ult
110       \fontfamily\ttsbstdefault
111     \else
112       \fontfamily\substdefault
113     \fi\fi\fi

```

Whatever default was chosen, we claim now (locally hopefully) that it can handle all slots (even if not true) to avoid looping in certain situations, e.g., when something was set up incorrectly.

```

114     \@namedef{TS1:\f@family}{0}%
115     \selectfont#1%
116   \egroup
117 }

```

`\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 a poor man’s Euro symbol by combining a “C” with a “=”.

```

118 \def\tc@fake@euro#1{%
119     \leavevmode
120     \@font@info{Faking \noexpand#1for font family
121                 \f@family\MessageBreak in TS1 encoding}%
122     \valign{##\cr
123         \vfil\hbox to 0.07em{\dimen@\f@size\p@
124                             \math@fontsfalse
125                             \fontsize{.7\dimen@}\z@\selectfont=\hss}%
126         \vfil\cr%
127         \hbox{C}\crrcr
128     }%
129 }

```

\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.

```
130 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
```

Accents and been mad an error in the `textcomp` package when not available. Now that we provide the functionality in the kernel we avoid the error by swapping in a T1 accent if the TS1 accent is not available.

```
131 %\def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
```

```
132 \def\tc@check@accent#1{\CheckEncodingSubset\UseTextAccent{TS1}{\tc@swap@accent#1}}
```

```
133 \def\tc@swap@accent#1#2{\UseTextAccent{T1}#1}
```

57 Sub-encodings

Here are the default definitions for the TS1 symbols. First those that we assume are always available if a font implements TS1.

```

134 \DeclareTextSymbolDefault{\textdollar}{TS1}
135 \UndeclareTextCommand{\textdollar} {OT1}           % don't use the OT1 def any longer
136 \DeclareTextSymbolDefault{\textsterling}{TS1}
137 \UndeclareTextCommand{\textsterling}{OT1}          % don't use the OT1 def any longer
138 \DeclareTextSymbolDefault{\textperthousand}{TS1}
139 \UndeclareTextCommand{\textperthousand}{T1}        % don't use the T1 def
140                                                     % should actually not be declaed
141                                                     % in the t1enc.def file instead
142                                                     % otherwise fontenc would overwrite
143                                                     % it again (legacy tests!)

```

These are the standard `itemize` and footnote symbols originally taken from OMS and now from TS1:

```

144 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
145 \DeclareTextSymbolDefault{\textbullet}{TS1}
146 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
147 \DeclareTextSymbolDefault{\textdagger}{TS1}
148 \DeclareTextSymbolDefault{\textparagraph}{TS1}

```

```

149 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
150 \DeclareTextSymbolDefault{\textsection}{TS1}

```

And here are the other TS1 glyphs that are implemented by every font (or nearly everyone—a few are commented out and moved to sub-encoding 9, because they aren’t around in one or two fonts.

```

151 %%\DeclareTextSymbolDefault{\textbardbl}{TS1} % subst in sub-enc 9 above
152 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
153 %%\DeclareTextSymbolDefault{\textcelsius}{TS1} % subst in sub-enc 9 above
154 \DeclareTextSymbolDefault{\textcent}{TS1}
155 \DeclareTextSymbolDefault{\textcopyright}{TS1}
156 \DeclareTextSymbolDefault{\textdegree}{TS1}
157 \DeclareTextSymbolDefault{\textdiv}{TS1}
158 \DeclareTextSymbolDefault{\textlnot}{TS1}
159 \DeclareTextSymbolDefault{\textonehalf}{TS1}
160 \DeclareTextSymbolDefault{\textonequarter}{TS1}
161 %%\DeclareTextSymbolDefault{\textonesuperior}{TS1} % subst in sub-enc 9 above
162 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
163 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
164 \DeclareTextSymbolDefault{\textpm}{TS1}
165 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
166 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
167 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
168 \DeclareTextSymbolDefault{\textregistered}{TS1}
169 %%\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1} % subst in sub-enc 9 above
170 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
171 %%\DeclareTextSymbolDefault{\textthreesuperior}{TS1} % subst in sub-enc 9 above
172 \DeclareTextSymbolDefault{\texttimes}{TS1}
173 \DeclareTextSymbolDefault{\texttrademark}{TS1}
174 %%\DeclareTextSymbolDefault{\texttwelveudash}{TS1} % subst in sub-enc 9 above
175 %%\DeclareTextSymbolDefault{\texttwosuperior}{TS1} % subst in sub-enc 9 above
176 \DeclareTextSymbolDefault{\textyen}{TS1}

177 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
178 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}

```

In the following sections the remaining default definitions are ordered by sub-encoding in which they are become unavailable (i.e., they are not provided in the sub-encoding with that number and all sub-encodings with higher numbers.

Thus the symbols that are available in sub-encoding x are the symbols above (always available) and the symbols list in the sections for sub-encodings $x + 1$ and higher.

57.1 Sub-encoding 1 (drop symbols not working in Latin Modern)

The `\textcircled` is available but the glyph is simply too small so we keep using the OMS glyph.

```

179 \DeclareTextCommandDefault{\textcircled}
180 {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OMS}}1\textcircled}

```

57.2 Sub-encoding 2 (majority of new OTF fonts via autotinst)

```

181 \DeclareTextCommandDefault{\t}
182 {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OML}}2\t}

```

Capital accents are really only very seldom implemented, so from sub-encoding 2 onwards we use the normal T1 accents if they are asked for in the document.

In Unicode engines we don't implement them at all but always use the basic accents instead. whether that works or not really depends on the font, something like `\X` usually comes out wrong in Unicode engines.

```

183 \ifx\Umathcode\@undefined
184 \DeclareTextCommandDefault{\capitalacute} {\tc@check@accent{'}2\capitalacute}
185 \DeclareTextCommandDefault{\capitalbreve} {\tc@check@accent{u}2\capitalbreve}
186 \DeclareTextCommandDefault{\capitalcaron} {\tc@check@accent{v}2\capitalcaron}
187 \DeclareTextCommandDefault{\capitalcedilla} {\tc@check@accent{c}2\capitalcedilla}
188 \DeclareTextCommandDefault{\capitalcircumflex} {\tc@check@accent{^}2\capitalcircumflex}
189 \DeclareTextCommandDefault{\capitaldieresis} {\tc@check@accent{"}2\capitaldieresis}
190 \DeclareTextCommandDefault{\capitaldotaccent} {\tc@check@accent{.}2\capitaldotaccent}
191 \DeclareTextCommandDefault{\capitalgrave} {\tc@check@accent{'}2\capitalgrave}
192 \DeclareTextCommandDefault{\capitalhungarumlaut} {\tc@check@accent{H}2\capitalhungarumlaut}
193 \DeclareTextCommandDefault{\capitalmacron} {\tc@check@accent{=}2\capitalmacron}
194 \DeclareTextCommandDefault{\capitalogonek} {\tc@check@accent{k}2\capitalogonek}
195 \DeclareTextCommandDefault{\capitalring} {\tc@check@accent{r}2\capitalring}
196 \DeclareTextCommandDefault{\capitaltie} {\tc@check@accent{t}2\capitaltie}
197 \DeclareTextCommandDefault{\capitaltilde} {\tc@check@accent{~}2\capitaltilde}

```

For `\newtie` and `\capitalnewtie` this is actually wrong, they should pick up the accent from the substitution font (not done yet).

```

198 \DeclareTextCommandDefault{\newtie} {\tc@check@accent{t}2\newtie}
199 \DeclareTextCommandDefault{\capitalnewtie} {\tc@check@accent{t}2\capitalnewtie}

```

In Unicode engines we just execute the simple accents:

```

200 \else
201 \def\capitalacute{'}
202 \def\capitalbreve{u}
203 \def\capitalcaron{v}
204 \def\capitalcedilla{c}
205 \def\capitalcircumflex{^}
206 \def\capitaldieresis{"}
207 \def\capitaldotaccent{.}
208 \def\capitalgrave{'}
209 \def\capitalhungarumlaut{H}
210 \def\capitalmacron{=}
211 \def\capitalnewtie{t}
212 \def\capitalogonek{k}
213 \def\capitalring{r}
214 \def\capitaltie{t}
215 \def\capitaltilde{~}
216 \def\newtie{t}
217 \fi

```

The next two symbols exist in some fonts (faked?), but we ignore that to keep the subsets reasonable compact and most important linear.

```

218 \DeclareTextCommandDefault{\textlbrackdbl} {\tc@check@symbol12\textlbrackdbl}
219 \DeclareTextCommandDefault{\textrbrackdbl} {\tc@check@symbol12\textrbrackdbl}

```

Old style numerals are again in some fonts but using `-OsF`, etc. is the better approach to get them, so we claim they aren't in sub-encoding 2 as that's true for most fonts.

```

220 \DeclareTextCommandDefault{\texteightoldstyle} {\tc@check@symbol2\texteightoldstyle}
221 \DeclareTextCommandDefault{\textfiveoldstyle} {\tc@check@symbol2\textfiveoldstyle}
222 \DeclareTextCommandDefault{\textfouroldstyle} {\tc@check@symbol2\textfouroldstyle}
223 \DeclareTextCommandDefault{\textnineoldstyle} {\tc@check@symbol2\textnineoldstyle}
224 \DeclareTextCommandDefault{\textoneoldstyle} {\tc@check@symbol2\textoneoldstyle}
225 \DeclareTextCommandDefault{\textsevenoldstyle} {\tc@check@symbol2\textsevenoldstyle}
226 \DeclareTextCommandDefault{\textsixoldstyle} {\tc@check@symbol2\textsixoldstyle}
227 \DeclareTextCommandDefault{\textthreeoldstyle} {\tc@check@symbol2\textthreeoldstyle}
228 \DeclareTextCommandDefault{\texttwooldstyle} {\tc@check@symbol2\texttwooldstyle}
229 \DeclareTextCommandDefault{\textzerooldstyle} {\tc@check@symbol2\textzerooldstyle}

```

The next set of glyphs is special to TeX fonts (and available with a few older PS fonts supported in the virtual fonts), but not any longer in the majority of fonts provided through autinst, so we pretend there aren't available in sub-encoding 2 and below.

```

230 \DeclareTextCommandDefault{\textacutedbl} {\tc@check@symbol2\textacutedbl}
231 \DeclareTextCommandDefault{\textasciiaacute} {\tc@check@symbol2\textasciiaacute}
232 \DeclareTextCommandDefault{\textasciibreve} {\tc@check@symbol2\textasciibreve}
233 \DeclareTextCommandDefault{\textasciicaron} {\tc@check@symbol2\textasciicaron}
234 \DeclareTextCommandDefault{\textasciidieresis} {\tc@check@symbol2\textasciidieresis}
235 \DeclareTextCommandDefault{\textasciigrave} {\tc@check@symbol2\textasciigrave}
236 \DeclareTextCommandDefault{\textasciimacron} {\tc@check@symbol2\textasciimacron}
237 \DeclareTextCommandDefault{\textgravedbl} {\tc@check@symbol2\textgravedbl}
238 \DeclareTextCommandDefault{\texttildelow} {\tc@check@symbol2\texttildelow}

```

Finally those below are only available in CM-based fonts but in no font that has its origin outside of the T_EX world.

```

239 \DeclareTextCommandDefault{\textbaht} {\tc@check@symbol2\textbaht}
240 \DeclareTextCommandDefault{\textbigcircle} {\tc@check@symbol2\textbigcircle}
241 \DeclareTextCommandDefault{\textborn} {\tc@check@symbol2\textborn}
242 \DeclareTextCommandDefault{\textcentoldstyle} {\tc@check@symbol2\textcentoldstyle}
243 \DeclareTextCommandDefault{\textcircledP} {\tc@check@symbol2\textcircledP}
244 \DeclareTextCommandDefault{\textcopleft} {\tc@check@symbol2\textcopleft}
245 \DeclareTextCommandDefault{\textdblhyphenchar} {\tc@check@symbol2\textdblhyphenchar}
246 \DeclareTextCommandDefault{\textdblhyphen} {\tc@check@symbol2\textdblhyphen}
247 \DeclareTextCommandDefault{\textdied} {\tc@check@symbol2\textdied}
248 \DeclareTextCommandDefault{\textdiscount} {\tc@check@symbol2\textdiscount}
249 \DeclareTextCommandDefault{\textdivorced} {\tc@check@symbol2\textdivorced}
250 \DeclareTextCommandDefault{\textdollaroldstyle} {\tc@check@symbol2\textdollaroldstyle}
251 \DeclareTextCommandDefault{\textguarani} {\tc@check@symbol2\textguarani}
252 \DeclareTextCommandDefault{\textleaf} {\tc@check@symbol2\textleaf}
253 \DeclareTextCommandDefault{\textlquill} {\tc@check@symbol2\textlquill}
254 \DeclareTextCommandDefault{\textmarried} {\tc@check@symbol2\textmarried}
255 \DeclareTextCommandDefault{\textmho} {\tc@check@symbol2\textmho}
256 \DeclareTextCommandDefault{\textmusicalnote} {\tc@check@symbol2\textmusicalnote}
257 \DeclareTextCommandDefault{\textnaira} {\tc@check@symbol2\textnaira}
258 \DeclareTextCommandDefault{\textopenbullet} {\tc@check@symbol2\textopenbullet}
259 \DeclareTextCommandDefault{\textpeso} {\tc@check@symbol2\textpeso}
260 \DeclareTextCommandDefault{\textpilcrow} {\tc@check@symbol2\textpilcrow}
261 \DeclareTextCommandDefault{\textrecipe} {\tc@check@symbol2\textrecipe}
262 \DeclareTextCommandDefault{\textreferencemark} {\tc@check@symbol2\textreferencemark}
263 \DeclareTextCommandDefault{\textrquill} {\tc@check@symbol2\textrquill}
264 \DeclareTextCommandDefault{\textservicemark} {\tc@check@symbol2\textservicemark}
265 \DeclareTextCommandDefault{\textsurd} {\tc@check@symbol2\textsurd}

```

The `\textpertenthousand` also belongs in this group but here we have a

choice: in T1 there is definition for `\textpertenthousand` making the symbol up from % and `\char 24` (twice) but in many fonts that char doesn't exist and the slot is reused for random ligatures. So better not use it because often it is wrong. But pointing to TS1 is also not great as only a few fonts have it as a real symbol, so we get a substitution to CM or LM.

Alternatively we could just state that the symbol is unavailable in those fonts. For now I substitute.

```
266 \DeclareTextCommandDefault{\textpertenthousand} {\tc@check@symbol2\textpertenthousand}
267 \UndeclareTextCommand{\textpertenthousand}{T1}
```

57.3 Sub-encoding 3

Sub-encoding 2 is the one where we loose many symbols. In the higher-numbered sub-encodings we see only a few dropped additionally.

```
268 \DeclareTextCommandDefault{\textlangle} {\tc@check@symbol3\textlangle}
269 \DeclareTextCommandDefault{\textrangle} {\tc@check@symbol3\textrangle}
```

57.4 Sub-encoding 4

```
270 \DeclareTextCommandDefault{\textcolonmonetary} {\tc@check@symbol4\textcolonmonetary}
271 \DeclareTextCommandDefault{\textdong} {\tc@check@symbol4\textdong}
272 \DeclareTextCommandDefault{\textdownarrow} {\tc@check@symbol4\textdownarrow}
273 \DeclareTextCommandDefault{\textleftarrow} {\tc@check@symbol4\textleftarrow}
274 \DeclareTextCommandDefault{\textlira} {\tc@check@symbol4\textlira}
275 \DeclareTextCommandDefault{\textrightarrow} {\tc@check@symbol4\textrightarrow}
276 \DeclareTextCommandDefault{\textuparrow} {\tc@check@symbol4\textuparrow}
277 \DeclareTextCommandDefault{\textwon} {\tc@check@symbol4\textwon}
```

57.5 Sub-encoding 5 (most older PS fonts)

Most older PS fonts (supported in T_EX since the early nineties when virtual fonts became available) are sorted under this sub-encoding. But in reality, many of them don't have all glyphs that should be available in sub-encoding 5. Instead they show little squares, i.e., they produce “tofu” if you are unlucky.

But the coverage is so random that it is impossible to sort them properly and if we tried to ensure that they only typeset those glyphs that are really always available wouput put them all into sub-encoding 9 so that's a compromise really.

Modern fonts that don't typeset a tofu character if a glyph is missing are only cataloged as sub-encoding 5 if they really support of its glyph set.

```
278 \DeclareTextCommandDefault{\textestimated} {\tc@check@symbol5\textestimated}
279 \DeclareTextCommandDefault{\textnumero} {\tc@check@symbol5\textnumero}
```

57.6 Sub-encoding 6

```
280 \DeclareTextCommandDefault{\textflorin} {\tc@check@symbol6\textflorin}
281 \DeclareTextCommandDefault{\textcurrency} {\tc@check@symbol6\textcurrency}
```

57.7 Sub-encoding 7

```
282 \DeclareTextCommandDefault{\textfractionsolidus}{\tc@check@symbol7\textfractionsolidus}
283 \DeclareTextCommandDefault{\textohm} {\tc@check@symbol7\textohm}
284 \DeclareTextCommandDefault{\textmu} {\tc@check@symbol7\textmu}
285 \DeclareTextCommandDefault{\textminus} {\tc@check@symbol7\textminus}
```


57.8 Sub-encoding 8

```
286 \DeclareTextCommandDefault{\textblank}{\tc@check@symbol{8}\textblank}
287 \DeclareTextCommandDefault{\textinterrobangdown}{\tc@check@symbol{8}\textinterrobangdown}
288 \DeclareTextCommandDefault{\textinterrobang}{\tc@check@symbol{8}\textinterrobang}
```

Fonts with this sub-encoding don't have a Euro symbol, but instead of substituting we fake it.

```
289 \DeclareTextCommandDefault{\texteuro}
290 {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro{8}\texteuro}
```

57.9 Sub-encoding 9 (most missing)

```
291 \DeclareTextCommandDefault{\textcelsius}{\tc@check@symbol{9}\textcelsius}
292 \DeclareTextCommandDefault{\textonesuperior}{\tc@check@symbol{9}\textonesuperior}
293 \DeclareTextCommandDefault{\textthreequartersemdash}{\tc@check@symbol{9}\textthreequartersemdash}
294 \DeclareTextCommandDefault{\textthreesuperior}{\tc@check@symbol{9}\textthreesuperior}
295 \DeclareTextCommandDefault{\texttwelveudash}{\tc@check@symbol{9}\texttwelveudash}
296 \DeclareTextCommandDefault{\texttwosuperior}{\tc@check@symbol{9}\texttwosuperior}
297 \DeclareTextCommandDefault{\textbardbl}{\tc@check@symbol{9}\textbardbl}
```

58 Unicode engine specials

If we are using a unicode engine we handle some glyphs differently, so this here are the definitions for the Unicode encoding (overwriting the defaults above).

```
298 \ifx \Umathcode\@undefined \else
```

This set should be taken from TS1 encoding even if it means you get it from the default font for that encoding.

```
299 %\DeclareTextSymbol{\textcopyright}{TS1}{171}
300 %\DeclareTextSymbol{\textdblhyphen}{TS1}{45}
301 %\DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}
302 %\DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
303 %\DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
304 %\DeclareTextSymbol{\textleaf}{TS1}{108}
305 %\DeclareTextSymbol{\texttwelveudash}{TS1}{21}
306 %\DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
```

If oldstyle numerals are asked for we just use \oldstylenums.

```
307 \DeclareTextCommand{\textzerooldstyle} \UnicodeEncodingName{\oldstylenums{0}}
308 \DeclareTextCommand{\textoneoldstyle} \UnicodeEncodingName{\oldstylenums{1}}
309 \DeclareTextCommand{\texttwooldstyle} \UnicodeEncodingName{\oldstylenums{2}}
310 \DeclareTextCommand{\textthreeoldstyle} \UnicodeEncodingName{\oldstylenums{3}}
311 \DeclareTextCommand{\textfouroldstyle} \UnicodeEncodingName{\oldstylenums{4}}
312 \DeclareTextCommand{\textfiveoldstyle} \UnicodeEncodingName{\oldstylenums{5}}
313 \DeclareTextCommand{\textsixoldstyle} \UnicodeEncodingName{\oldstylenums{6}}
314 \DeclareTextCommand{\textsevenoldstyle} \UnicodeEncodingName{\oldstylenums{7}}
315 \DeclareTextCommand{\texteightoldstyle} \UnicodeEncodingName{\oldstylenums{8}}
316 \DeclareTextCommand{\textnineoldstyle} \UnicodeEncodingName{\oldstylenums{9}}
```

These have Unicode slots so this should be integrated into TU explicitly

```
317 \DeclareTextSymbol{\textpilcrow} \UnicodeEncodingName{"00B6}
318 \DeclareTextSymbol{\textborn} \UnicodeEncodingName{"002A}
319 \DeclareTextSymbol{\textdied} \UnicodeEncodingName{"2020}
320 \DeclareTextSymbol{\textlbrackdbl} \UnicodeEncodingName{"27E6}
321 \DeclareTextSymbol{\textrbrackdbl} \UnicodeEncodingName{"27E7}
322 \DeclareTextSymbol{\textguarani} \UnicodeEncodingName{"20B2}
```

We could make `\textcentoldstyle` and `\textdollaroldstyle` point to dollar and cent in the Unicode encoding

```
323 %\DeclareTextSymbol{\textcentoldstyle}          \UnicodeEncodingName{"00A2}
324 %\DeclareTextSymbol{\textdollaroldstyle}        \UnicodeEncodingName{"0024}
```

but I think it is better to pick them up from TS1 even if that usually means LMR fonts

```
325 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
326 \DeclareTextSymbol{\textcentoldstyle} {TS1}{139}
327 \fi                                     % --- END of Unicode engines specials
```

59 Font family sub-encodings setup

We declare the subsets for a good number of fonts in the kernel ...

But first the default for anything that is not declared. We use 9 which is most like much too conservative but with the advantage that we aren't getting missing glyphs (or at least that this very very unlikely). For nearly all font in the T_EX Live distribution of 2019 "correct" classifications are given below, so that this default is only used for new font families, and over time the right classifications can be added here too.

```
328 \DeclareEncodingSubset{TS1}{?}{9}
```

This first block contains the fonts that have been already supported by the `textcomp` package way back, i.e., the font families that have T_EX support since the mid-nineties.

```
329 \DeclareEncodingSubset{TS1}{ccr}      {0}
330 \DeclareEncodingSubset{TS1}{cmbr}     {0}
331 \DeclareEncodingSubset{TS1}{cmr}      {0}
332 \DeclareEncodingSubset{TS1}{cmss}     {0}
333 \DeclareEncodingSubset{TS1}{cmtl}     {0}
334 \DeclareEncodingSubset{TS1}{cmtt}     {0}
335 \DeclareEncodingSubset{TS1}{cmvtt}    {0}
336 \DeclareEncodingSubset{TS1}{pxr}      {0}
337 \DeclareEncodingSubset{TS1}{pxss}     {0}
338 \DeclareEncodingSubset{TS1}{pxtt}     {0}
339 \DeclareEncodingSubset{TS1}{qag}      {0}
340 \DeclareEncodingSubset{TS1}{qbk}      {0}
341 \DeclareEncodingSubset{TS1}{qcr}      {0}
342 \DeclareEncodingSubset{TS1}{qcs}      {0}
343 \DeclareEncodingSubset{TS1}{qhvc}     {0}
344 \DeclareEncodingSubset{TS1}{qhv}      {0}
345 \DeclareEncodingSubset{TS1}{qpl}      {0}
346 \DeclareEncodingSubset{TS1}{qtm}      {0}
347 \DeclareEncodingSubset{TS1}{qzc}      {0}
348 \DeclareEncodingSubset{TS1}{txr}      {0}
349 \DeclareEncodingSubset{TS1}{txss}     {0}
350 \DeclareEncodingSubset{TS1}{txtt}     {0}

351 \DeclareEncodingSubset{TS1}{lmr}      {1}
352 \DeclareEncodingSubset{TS1}{lmdh}     {1}
353 \DeclareEncodingSubset{TS1}{lmss}     {1}
354 \DeclareEncodingSubset{TS1}{lmssq}    {1}
355 \DeclareEncodingSubset{TS1}{lmvtt}    {1}
```

```

356 \DeclareEncodingSubset{TS1}{lmtt}    {1} % missing TM, SM, pertenthousand for some reason
357 \DeclareEncodingSubset{TS1}{ptmx}    {2}
358 \DeclareEncodingSubset{TS1}{ptmj}    {2}
359 \DeclareEncodingSubset{TS1}{ul8}     {2}

360 \DeclareEncodingSubset{TS1}{bch}     {5} % tofu for blank, ohm
361 \DeclareEncodingSubset{TS1}{futj}    {5} % tofu for blank, interrobang/down, ohm
362 \DeclareEncodingSubset{TS1}{futs}    {5} % tofu for blank, ohm
363 \DeclareEncodingSubset{TS1}{futex}   {5} % probably (currently broken distrib)
364 \DeclareEncodingSubset{TS1}{pag}     {5} % tofu for blank, interrobang/down, ohm
365 \DeclareEncodingSubset{TS1}{pbk}     {5} % tofu for blank, interrobang/down, ohm
366 \DeclareEncodingSubset{TS1}{pcr}     {5} % tofu for blank, interrobang/down, ohm
367 \DeclareEncodingSubset{TS1}{phv}     {5} % tofu for blank, interrobang/down, ohm
368 \DeclareEncodingSubset{TS1}{pnc}     {5} % tofu for blank, interrobang/down, ohm
369 \DeclareEncodingSubset{TS1}{pplj}    {5} % tofu for blank
370 \DeclareEncodingSubset{TS1}{pplx}    {5} % tofu for blank
371 \DeclareEncodingSubset{TS1}{ppl}     {5} % tofu for blank interrobang/down
372 \DeclareEncodingSubset{TS1}{ptm}     {5} % tofu for blank, interrobang/down, ohm
373 \DeclareEncodingSubset{TS1}{pzc}     {5} % tofu for blank, interrobang/down, ohm
374 \DeclareEncodingSubset{TS1}{ul9}     {5} % tofu for blank, interrobang/down, ohm

375 \DeclareEncodingSubset{TS1}{dayroms} {6} % tofu for blank, interrobang/down, ohm
376 \DeclareEncodingSubset{TS1}{dayrom}  {6} % tofu for blank, interrobang/down, ohm

377 \DeclareEncodingSubset{TS1}{augie}   {8} % really only missing euro
378 \DeclareEncodingSubset{TS1}{put}     {8}
379 \DeclareEncodingSubset{TS1}{uag}     {8} % probably (currently broken distrib)
380 \DeclareEncodingSubset{TS1}{ugq}     {8}

381 \DeclareEncodingSubset{TS1}{zi4}     {9}

```

LucidaBright (sold through TUG) probably not quite correct, I guess as I have the older fonts ...

```

382 \DeclareEncodingSubset{TS1}{hls}     {5}
383 \DeclareEncodingSubset{TS1}{hlst}    {5}
384 \DeclareEncodingSubset{TS1}{hlct}    {5}
385 \DeclareEncodingSubset{TS1}{hlh}     {5}
386 \DeclareEncodingSubset{TS1}{hlx}     {8}
387 \DeclareEncodingSubset{TS1}{hlce}    {8}
388 \DeclareEncodingSubset{TS1}{hlcn}    {8}
389 \DeclareEncodingSubset{TS1}{hlcw}    {8}
390 \DeclareEncodingSubset{TS1}{hlcf}    {8}

```

Below are the newer fonts that have support files for L^AT_EX. With very few exceptions the classifications are done so that that all characters are correctly produced (either being available in the font or substituted).

There are a few fonts that contain “tofu” squares in places (instead of a real glyph) and in a few cases some really seldom needed chars are unavailable, i.e., produce missing glyphs (to avoid that a large number of available chars are unnecessarily substituted).

```

391 \DeclareEncodingSubset{TS1}{lato-*}   {0} % with a bunch of tofu inside
392 \DeclareEncodingSubset{TS1}{opensans-*} {0} % with a bunch of tofu inside
393 \DeclareEncodingSubset{TS1}{cantarell-*} {0} % with a bunch of tofu inside
394 \DeclareEncodingSubset{TS1}{fbb-*}     {0} % missing centoldstyle
395 \DeclareEncodingSubset{TS1}{tli}      {1} % with lots of tofu inside

```

```

396 \DeclareEncodingSubset{TS1}{Alegreya-*} {2}
397 \DeclareEncodingSubset{TS1}{AlegreyaSans-*} {2}
398 \DeclareEncodingSubset{TS1}{DejaVuSans-TLF} {2}
399 \DeclareEncodingSubset{TS1}{DejaVuSansCondensed-TLF} {2}
400 \DeclareEncodingSubset{TS1}{DejaVuSansMono-TLF} {2}
401 \DeclareEncodingSubset{TS1}{EBGaramond-*} {2}
402 \DeclareEncodingSubset{TS1}{Tempora-TLF} {2}
403 \DeclareEncodingSubset{TS1}{Tempora-TOf} {2}

404 \DeclareEncodingSubset{TS1}{Arimo-TLF} {3}
405 \DeclareEncodingSubset{TS1}{Carlito-*} {3}
406 \DeclareEncodingSubset{TS1}{FiraSans-*} {3}
407 \DeclareEncodingSubset{TS1}{IBMPlexSans-TLF} {3}
408 \DeclareEncodingSubset{TS1}{Merriweather-Of} {3}
409 \DeclareEncodingSubset{TS1}{Montserrat-*} {3}
410 \DeclareEncodingSubset{TS1}{MontserratAlternates-*} {3}
411 \DeclareEncodingSubset{TS1}{SourceCodePro-TLF} {3}
412 \DeclareEncodingSubset{TS1}{SourceCodePro-TOf} {3}
413 \DeclareEncodingSubset{TS1}{SourceSansPro-*} {3}
414 \DeclareEncodingSubset{TS1}{SourceSerifPro-*} {3}
415 \DeclareEncodingSubset{TS1}{Tinos-TLF} {3}

416 \DeclareEncodingSubset{TS1}{AccanthisADFStdNoThree-LF} {4}
417 \DeclareEncodingSubset{TS1}{Cabin-TLF} {4}
418 \DeclareEncodingSubset{TS1}{Caladea-TLF} {4}
419 \DeclareEncodingSubset{TS1}{Chivo-*} {4}
420 \DeclareEncodingSubset{TS1}{ClearSans-TLF} {4}
421 \DeclareEncodingSubset{TS1}{Coelacanth-LF} {4}
422 \DeclareEncodingSubset{TS1}{CrimsonPro-*} {4}
423 \DeclareEncodingSubset{TS1}{FiraMono-TLF} {4}
424 \DeclareEncodingSubset{TS1}{FiraMono-TOf} {4}
425 \DeclareEncodingSubset{TS1}{Go-TLF} {4}
426 \DeclareEncodingSubset{TS1}{GoMono-TLF} {4}
427 \DeclareEncodingSubset{TS1}{InriaSans-*} {4}
428 \DeclareEncodingSubset{TS1}{InriaSerif-*} {4}
429 \DeclareEncodingSubset{TS1}{LibertinusSans-*} {4}
430 \DeclareEncodingSubset{TS1}{LibertinusSerif-*} {4}
431 \DeclareEncodingSubset{TS1}{LibreBodoni-TLF} {4}
432 \DeclareEncodingSubset{TS1}{LibreFranklin-TLF} {4}
433 \DeclareEncodingSubset{TS1}{LinguisticsPro-LF} {4}
434 \DeclareEncodingSubset{TS1}{LinguisticsPro-Of} {4}
435 \DeclareEncodingSubset{TS1}{LinuxBiolumT-*} {4}
436 \DeclareEncodingSubset{TS1}{LinuxLibertineT-*} {4}
437 \DeclareEncodingSubset{TS1}{MerriweatherSans-Of} {4}
438 \DeclareEncodingSubset{TS1}{MintSpirit-*} {4}
439 \DeclareEncodingSubset{TS1}{MintSpiritNoTwo-*} {4}
440 \DeclareEncodingSubset{TS1}{PTMono-TLF} {4}
441 \DeclareEncodingSubset{TS1}{PTSans-TLF} {4}
442 \DeclareEncodingSubset{TS1}{PTSansCaption-TLF} {4}
443 \DeclareEncodingSubset{TS1}{PTSansNarrow-TLF} {4}
444 \DeclareEncodingSubset{TS1}{PTSerif-TLF} {4}
445 \DeclareEncodingSubset{TS1}{PTSerifCaption-TLF} {4}
446 \DeclareEncodingSubset{TS1}{Raleway-TLF} {4}
447 \DeclareEncodingSubset{TS1}{Raleway-TOf} {4}
448 \DeclareEncodingSubset{TS1}{Roboto-*} {4}

```

```

449 \DeclareEncodingSubset{TS1}{RobotoMono-TLF} {4}
450 \DeclareEncodingSubset{TS1}{RobotoSlab-TLF} {4}
451 \DeclareEncodingSubset{TS1}{Rosario-*} {4}
452 \DeclareEncodingSubset{TS1}{SticksTooText-*} {4}
453 \DeclareEncodingSubset{TS1}{UniversalisADFStd-LF} {4}

454 \DeclareEncodingSubset{TS1}{Almendra-OfF} {5}
455 \DeclareEncodingSubset{TS1}{Baskervaldx-*} {5}
456 \DeclareEncodingSubset{TS1}{BaskervilleF-*} {5}
457 \DeclareEncodingSubset{TS1}{Bitter-TLF} {5}
458 \DeclareEncodingSubset{TS1}{Cinzel-LF} {5}
459 \DeclareEncodingSubset{TS1}{CinzelDecorative-LF} {5}
460 \DeclareEncodingSubset{TS1}{DejaVuSerif-TLF} {5}
461 \DeclareEncodingSubset{TS1}{DejaVuSerifCondensed-TLF} {5}
462 \DeclareEncodingSubset{TS1}{GilliusADF-LF} {5}
463 \DeclareEncodingSubset{TS1}{GilliusADFCond-LF} {5}
464 \DeclareEncodingSubset{TS1}{GilliusADFNoTwo-LF} {5}
465 \DeclareEncodingSubset{TS1}{GilliusADFNoTwoCond-LF} {5}
466 \DeclareEncodingSubset{TS1}{LobsterTwo-LF} {5}
467 \DeclareEncodingSubset{TS1}{OldStandard-TLF} {5}
468 \DeclareEncodingSubset{TS1}{PlayfairDisplay-TLF} {5}
469 \DeclareEncodingSubset{TS1}{PlayfairDisplay-TOfF} {5}
470 \DeclareEncodingSubset{TS1}{TheanoDidot-TLF} {5}
471 \DeclareEncodingSubset{TS1}{TheanoDidot-TOfF} {5}
472 \DeclareEncodingSubset{TS1}{TheanoModern-TLF} {5}
473 \DeclareEncodingSubset{TS1}{TheanoModern-TOfF} {5}
474 \DeclareEncodingSubset{TS1}{TheanoOldStyle-TLF} {5}
475 \DeclareEncodingSubset{TS1}{TheanoOldStyle-TOfF} {5}

476 \DeclareEncodingSubset{TS1}{Crimson-TLF} {6}
477 \DeclareEncodingSubset{TS1}{IBMPlexMono-TLF} {6}
478 \DeclareEncodingSubset{TS1}{IBMPlexSerif-TLF} {6}
479 \DeclareEncodingSubset{TS1}{LibertinusMono-TLF} {6}
480 \DeclareEncodingSubset{TS1}{LibertinusSerifDisplay-LF} {6}
481 \DeclareEncodingSubset{TS1}{LinuxLibertineDisplayT-*} {6}
482 \DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-LF} {6}
483 \DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-TLF} {6}
484 \DeclareEncodingSubset{TS1}{Overlock-LF} {6}

485 \DeclareEncodingSubset{TS1}{CormorantGaramond-*} {7}
486 \DeclareEncodingSubset{TS1}{Heuristica-TLF} {7}
487 \DeclareEncodingSubset{TS1}{Heuristica-TOfF} {7}
488 \DeclareEncodingSubset{TS1}{IMFELLEnglish-TLF} {7}
489 \DeclareEncodingSubset{TS1}{LibreBaskerville-TLF} {7}
490 \DeclareEncodingSubset{TS1}{LibreCaslon-*} {7}
491 \DeclareEncodingSubset{TS1}{Marcellus-LF} {7}
492 \DeclareEncodingSubset{TS1}{NotoSans-*} {7}
493 \DeclareEncodingSubset{TS1}{NotoSansMono-TLF} {7}
494 \DeclareEncodingSubset{TS1}{NotoSansMono-TOfF} {7}
495 \DeclareEncodingSubset{TS1}{NotoSerif-*} {7}
496 \DeclareEncodingSubset{TS1}{Quattrocento-TLF} {7}
497 \DeclareEncodingSubset{TS1}{QuattrocentoSans-TLF} {7}
498 \DeclareEncodingSubset{TS1}{XCharter-TLF} {7}
499 \DeclareEncodingSubset{TS1}{XCharter-TOfF} {7}
500 \DeclareEncodingSubset{TS1}{erewhon-*} {7}
501 \DeclareEncodingSubset{TS1}{ComicNeue-TLF} {7}

```

```

502 \DeclareEncodingSubset{TS1}{ComicNeueAngular-TLF}      {7}
503 \DeclareEncodingSubset{TS1}{Forum-LF}                   {7}  % the superiors are missing
504 \DeclareEncodingSubset{TS1}{Cochineal-*}                 {8}
505 \DeclareEncodingSubset{TS1}{AlgolRevived-TLF}           {9}

```

60 Legacy symbol support for lists and footnote symbols

`\UseLegacyTextSymbols`

```

506 \def\UseLegacyTextSymbols{%
507   \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}%
508   \DeclareTextSymbolDefault{\textbardbl}{OMS}%
509   \DeclareTextSymbolDefault{\textbullet}{OMS}%
510   \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}%
511   \DeclareTextSymbolDefault{\textdagger}{OMS}%
512   \DeclareTextSymbolDefault{\textparagraph}{OMS}%
513   \DeclareTextSymbolDefault{\textperiodcentered}{OMS}%
514   \DeclareTextSymbolDefault{\textsection}{OMS}%
515   \UndeclareTextCommand{\textsection}{T1}%
516   \expandafter\let\csname oldstylenums \expandafter\endcsname
517     \csname legacyoldstylenums \endcsname
518 }

```

`\textlegacyasteriskcentered` Here are new names for the legacy symbols that L^AT_EX used to pick up from the OMS encoded fonts (and used for itemize lists or footnote symbols).

`\textlegacybardbl` We go the roundabout way via separate OMS declarations so that

`\textlegacybullet` `\renewcommand\textbullet{\textlegacybullet}`

`\textlegacydaggerdbl` doesn't produce an endless loop.

```

\textlegacydagger
\textlegacyparagraph
\textlegacyperiodcentered
\textlegacysection
519 \DeclareTextSymbol{\textlegacyasteriskcentered}{OMS}{3}  % "03
520 \DeclareTextSymbol{\textlegacybardbl}{OMS}{107}          % "6B
521 \DeclareTextSymbol{\textlegacybullet}{OMS}{15}           % "0F
522 \DeclareTextSymbol{\textlegacydaggerdbl}{OMS}{122}       % "7A
523 \DeclareTextSymbol{\textlegacydagger}{OMS}{121}          % "79
524 \DeclareTextSymbol{\textlegacyparagraph}{OMS}{123}       % "7B
525 \DeclareTextSymbol{\textlegacyperiodcentered}{OMS}{1}    % "01
526 \DeclareTextSymbol{\textlegacysection}{OMS}{120}         % "78

527 \DeclareTextSymbolDefault{\textlegacyasteriskcentered}{OMS}
528 \DeclareTextSymbolDefault{\textlegacybardbl}{OMS}
529 \DeclareTextSymbolDefault{\textlegacybullet}{OMS}
530 \DeclareTextSymbolDefault{\textlegacydaggerdbl}{OMS}
531 \DeclareTextSymbolDefault{\textlegacydagger}{OMS}
532 \DeclareTextSymbolDefault{\textlegacyparagraph}{OMS}
533 \DeclareTextSymbolDefault{\textlegacyperiodcentered}{OMS}
534 \DeclareTextSymbolDefault{\textlegacysection}{OMS}

```

Supporting rollback ...

```

535 </2ekernel | latexrelease>
536 <latexrelease>\EndIncludeInRelease
537 <latexrelease>\IncludeInRelease{0000/00/00}%

```

```

538 \latexrelease \DeclareEncodingSubset\Text companion symbols}%
539 \latexrelease
540 \latexrelease \let\DeclareEncodingSubset\@undefined
541 \latexrelease \let\CheckEncodingSubset\@undefined
542 \latexrelease
543 \latexrelease \DeclareTextSymbolDefault{\textdollar}{OT1}
544 \latexrelease \DeclareTextSymbolDefault{\textsterling}{OT1}
545 \latexrelease \DeclareTextCommand{\textdollar}{OT1}{\hmode\bgroup
546 \latexrelease \ifdim \fontdimen\@ne\font >\z@
547 \latexrelease \slshape
548 \latexrelease \else
549 \latexrelease \upshape
550 \latexrelease \fi
551 \latexrelease \char'\$ \egroup}
552 \latexrelease \DeclareTextCommand{\textsterling}{OT1}{\hmode\bgroup
553 \latexrelease \ifdim \fontdimen\@ne\font >\z@
554 \latexrelease \itshape
555 \latexrelease \else
556 \latexrelease \fontshape{ui}\selectfont
557 \latexrelease \fi
558 \latexrelease \char'\$ \egroup}
559 \latexrelease \DeclareTextCommand{\textperthousand}{T1}
560 \latexrelease {\%\char 24 }
561 \latexrelease
562 \latexrelease \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
563 \latexrelease \DeclareTextSymbolDefault{\textbullet}{OMS}
564 \latexrelease \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
565 \latexrelease \DeclareTextSymbolDefault{\textdagger}{OMS}
566 \latexrelease \DeclareTextSymbolDefault{\textparagraph}{OMS}
567 \latexrelease \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
568 \latexrelease \DeclareTextSymbolDefault{\textsection}{OMS}
569 \latexrelease
570 \latexrelease \DeclareTextSymbolDefault{\textbardbl}{OMS}
571 \latexrelease \let\textbrokenbar\@undefined
572 \latexrelease \let\textcelsius\@undefined
573 \latexrelease \let\textcent\@undefined
574 \latexrelease \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
575 \latexrelease \let\textdegree\@undefined
576 \latexrelease \let\textdiv\@undefined
577 \latexrelease \let\textlnot\@undefined
578 \latexrelease \let\textonehalf\@undefined
579 \latexrelease \let\textonequarter\@undefined
580 \latexrelease \let\textonesuperior\@undefined
581 \latexrelease \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
582 \latexrelease \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}
583 \latexrelease \let\textpm\@undefined
584 \latexrelease \let\textquotesingle\@undefined
585 \latexrelease \let\textquotestraightbase\@undefined
586 \latexrelease \let\textquotestraightdblbase\@undefined
587 \latexrelease \DeclareTextCommandDefault{\textregistered}{\textcircled{%
588 \latexrelease \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
589 \latexrelease \let\textthreequartersemdash\@undefined
590 \latexrelease \let\textthreequarters\@undefined
591 \latexrelease \let\textthreesuperior\@undefined

```

```

592 \let\texttimes\@undefined
593 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
594 \let\texttwelveudash\@undefined
595 \let\texttwosuperior\@undefined
596 \let\textyen\@undefined
597 \let\textasciicircled\@undefined
598 \let\textasciitilde\@undefined
599 \let\textascendercompwordmark\@undefined
600 \let\textasciicircled\@undefined
601 \DeclareTextAccentDefault{\textcircled}{OMS}
602 \DeclareTextAccentDefault{\textcircled}{OML}
603 \let\textasciicircled\@undefined
604 \let\textasciicircled\@undefined
605 \let\textasciicircled\@undefined
606 \let\textasciicircled\@undefined
607 \let\textasciicircled\@undefined
608 \let\textasciicircled\@undefined
609 \let\textasciicircled\@undefined
610 \let\textasciicircled\@undefined
611 \let\textasciicircled\@undefined
612 \let\textasciicircled\@undefined
613 \let\textasciicircled\@undefined
614 \let\textasciicircled\@undefined
615 \let\textasciicircled\@undefined
616 \let\textasciicircled\@undefined
617 \let\textasciicircled\@undefined
618 \let\textasciicircled\@undefined
619 \let\textasciicircled\@undefined
620 \let\textasciicircled\@undefined
621 \let\textasciicircled\@undefined
622 \let\textasciicircled\@undefined
623 \let\textasciicircled\@undefined
624 \let\textasciicircled\@undefined
625 \let\textasciicircled\@undefined
626 \let\textasciicircled\@undefined
627 \let\textasciicircled\@undefined
628 \let\textasciicircled\@undefined
629 \let\textasciicircled\@undefined
630 \let\textasciicircled\@undefined
631 \let\textasciicircled\@undefined
632 \let\textasciicircled\@undefined
633 \let\textasciicircled\@undefined
634 \let\textasciicircled\@undefined
635 \let\textasciicircled\@undefined
636 \let\textasciicircled\@undefined
637 \let\textasciicircled\@undefined
638 \let\textasciicircled\@undefined
639 \let\textasciicircled\@undefined
640 \let\textasciicircled\@undefined
641 \let\textasciicircled\@undefined
642 \let\textasciicircled\@undefined
643 \let\textasciicircled\@undefined
644 \let\textasciicircled\@undefined
645 \let\textasciicircled\@undefined

```



```

646 \let\textbigcircle\@undefined
647 \let\textborn\@undefined
648 \let\textcentoldstyle\@undefined
649 \let\textcircledP\@undefined
650 \let\textcopyleft\@undefined
651 \let\textdblhyphenchar\@undefined
652 \let\textdblhyphen\@undefined
653 \let\textdied\@undefined
654 \let\textdiscount\@undefined
655 \let\textdivorced\@undefined
656 \let\textdollaroldstyle\@undefined
657 \let\textguarani\@undefined
658 \let\textleaf\@undefined
659 \let\textlquill\@undefined
660 \let\textmarried\@undefined
661 \let\textmho\@undefined
662 \let\textmusicalnote\@undefined
663 \let\textnaira\@undefined
664 \let\textopenbullet\@undefined
665 \let\textpeso\@undefined
666 \let\textpilcrow\@undefined
667 \let\textrecipe\@undefined
668 \let\textreferencemark\@undefined
669 \let\texttrquill\@undefined
670 \let\textservicemark\@undefined
671 \let\textsurd\@undefined
672 \let\textrelease\@undefined
673 \DeclareTextCommand{\textpertenthousand}{T1}
674 \let\textrelease\@undefined
675 \let\textrelease\@undefined
676 \let\textlangle\@undefined
677 \let\textrangle\@undefined
678 \let\textrelease\@undefined
679 \let\textcolonmonetary\@undefined
680 \let\textdong\@undefined
681 \let\textdownarrow\@undefined
682 \let\textleftarrow\@undefined
683 \let\textlira\@undefined
684 \let\textrightarrow\@undefined
685 \let\textuparrow\@undefined
686 \let\textwon\@undefined
687 \let\textrelease\@undefined
688 \let\textestimated\@undefined
689 \let\textnumero\@undefined
690 \let\textrelease\@undefined
691 \let\textflorin\@undefined
692 \let\textcurrency\@undefined
693 \let\textrelease\@undefined
694 \let\textfractionsolidus\@undefined
695 \let\textohm\@undefined
696 \let\textmu\@undefined
697 \let\textminus\@undefined
698 \let\textrelease\@undefined
699 \let\textblank\@undefined

```

```

700 \let\textinterrobangdown\@undefined
701 \let\textinterrobang\@undefined
702 \let\textinterrobangdown\@undefined
703 \let\texteuro\@undefined
704 \let\textcelsius\@undefined
705 \let\textonesuperior\@undefined
706 \let\textthreequartersemdash\@undefined
707 \let\textthreesuperior\@undefined
708 \let\texttwelveudash\@undefined
709 \let\texttwosuperior\@undefined
710 \let\textbardbl\@undefined
711 \let\textbardbl\@undefined
712 \let\textbardbl\@undefined
713 \let\UseLegacyTextSymbols\@undefined
714 \let\textlegacyasteriskcentered\@undefined
715 \let\textlegacybardbl\@undefined
716 \let\textlegacybullet\@undefined
717 \let\textlegacydaggerdbl\@undefined
718 \let\textlegacydagger\@undefined
719 \let\textlegacyparagraph\@undefined
720 \let\textlegacyperiodcentered\@undefined
721 \let\textlegacysection\@undefined
722 \let\textlegacysection\@undefined
723 \EndIncludeInRelease
724 \*2ekernel
725 /2ekernel

```

61 The textcomp package

```

726 \*TS1sty
727 \NeedsTeXFormat{LaTeX2e}[1995/06/01]
728 \providecommand\DeclareRelease[3]{}
729 \providecommand\DeclareCurrentRelease[2]{}
730
731 \DeclareRelease{}{2018-08-11}{textcomp-2018-08-11.sty}
732 \DeclareCurrentRelease{}{2020-02-02}
733
734 \ProvidesPackage{textcomp}
735 [2020/02/02 v2.0m Standard LaTeX package]

```

This is implemented by defining the default subset:

```

736 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
737 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
738 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{8}}
739 \DeclareOption{safe}{\DeclareEncodingSubset{TS1}{?}{9}}

```

The default is set up in the kernel is “safe” these days for unknown fonts but LaTeX has definitions for most families so it seldom applies.

If a different default is used then one needs to check the results to ensure that there aren’t “missing glyphs”.

The next set of options define the warning level (default in the kernel is info only). If you load the package then the default changes to produce an error message if any substitution happens.

```

740 \def\tc@errorwarn{\PackageError{textcomp}}

```

```

741 \DeclareOption{error}{\gdef\tc@errorwarn{\PackageError{textcomp}}}
742 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2{\PackageWarning{textcomp}{#1}}}
743 \DeclareOption{quiet}{\gdef\tc@errorwarn#1#2{}}

```

The “force” option basically changes the sub-encoding to that of the default (which, unless changes, is 9 these days), i.e., it no longer depends on the font in use. This is mainly there because it might have been used in older documents, but not something that is recommended.

```

744 \DeclareOption{force}{%
745     \def\CheckEncodingSubset#1#2#3#4#5{%
746         \ifnum #4>%
747             0\csname #2:?\endcsname
748             \relax
749         \expandafter\@firstoftwo
750     \else
751         \expandafter\@secondoftwo
752     \fi
753     {#1{#2}}{#3}%
754     #5}%
755 }
756 \ExecuteOptions{}
757 \ProcessOptions\relax

```

There is not much else to do nowadays, because everything is already set up in the L^AT_EX kernel.

```

758 \InputIfFileExists{textcomp.cfg}
759 {\PackageInfo{textcomp}{Local configuration file used}}{}
760 </TS1sty>

```

61.1 The old textcomp package code

This section contains the old code for the textcomp package and its documentation. It is only used if we roll back prior to 2020. Thus all the rest is mainly for historians. Note that the old code categorised in the sub-encodings only into 6 classes not 10.

```

761 (*TS1oldsty)
762 \ProvidesPackage{textcomp}
763 [2018/08/11 v2.0j Standard LaTeX 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.

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-T_EX 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:

```

764 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
765   \space\space 5 = only ISO-Adobe without
766                       \string\textcurrency\MessageBreak
767   \space\space 4 = 5 + \string\texteuro\MessageBreak
768   \space\space 3 = 4 + \string\textohm\MessageBreak
769   \space\space 2 = 3 + \noexpand\textestimated+
770                       \string\textcurrency\MessageBreak
771   \space\space 1 = TS1 - \noexpand\textcircled-
772                       \string\t\MessageBreak
773   \space\space 0 = TS1 (full)\MessageBreak
774   Font families with sub-encoding setting implement\MessageBreak
775   only a restricted character set as indicated.\MessageBreak
776   Family '?' is the default used for unknown fonts.\MessageBreak
777   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 **?**.

```

778 \def\DeclareEncodingSubset#1#2#3{%
779   \@ifundefined{#1:#2}%
780     {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
781     {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
782   \@namedef{#1:#2}{#3}}
783 \@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

```
784 \newif\iftc@forced \tc@forcedfalse
```

This is implemented by defining the default subset:

```
785 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
786 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
787 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
788 \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.

```
789 \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.

```
790 \def\tc@errorwarn{\PackageError}
791 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
792 \DeclareOption{quiet}{\gdef\tc@errorwarn#1#2#3{}}
793 \ExecuteOptions{almostfull}
794 \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.

```
795 \iftc@forced
```

If the “force” option was given we always use the default for testing against.

```
796 \def\CheckEncodingSubset#1#2#3#4#5{%
797   \ifnum #4>%
798     0\csname #2:?\endcsname
```

```

799      \relax
800      \expandafter\@firstoftwo
801      \else
802      \expandafter\@secondoftwo
803  \fi
804  {\#1{\#2}}{\#3}%
805  #5%
806 }

```

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.

```

807 \else
808 \def\CheckEncodingSubset#1#2#3#4#5{%
809     \ifnum #4>%
810         \expandafter\ifx\csname #2:\f@family\endcsname\relax
811             0\csname #2:\f@family\endcsname
812         \else
813             \csname #2:\f@family\endcsname
814         \fi
815     \relax
816     \expandafter\@firstoftwo
817     \else
818     \expandafter\@secondoftwo
819 \fi
820 {\#1{\#2}}{\#3}%
821 #5%
822 }
823 \fi

```

$\backslash tc@subst$

```

824 \def\tc@subst#1{%
825     \tc@errorwarn{textcomp}%
826     {Symbol \string#1 not provided by\MessageBreak
827     font family \f@family\space
828     in TS1 encoding.\MessageBreak Default family used instead}\@eha
829     \bgroup\fontfamily\textcompfont\selectfont#1\egroup
830 }

```

$\backslash tc@error$ $\backslash tc@error$ is going to be used in arg #3 of $\backslash CheckEncodingSubset$ 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.

```

831 % error commands take argument:
832 % #1 symbol to be used
833 \def\tc@error#1{%
834     \PackageError{textcomp}% % should be latex error if general
835     {Accent \string#1 not provided by\MessageBreak
836     font family \f@family\space
837     in TS1 encoding}\@eha
838 }

```

$\backslash tc@fake@euro$ $\backslash tc@fake@euro$ is an example of a “fake” definition to use in arg #3 of $\backslash CheckEncodingSubset$ when a symbol is not available in a certain font family. Here we produce an Euro symbol by combining a “C” with a “=”.

```

839 \def\tc@fake@euro#1{%
840   \leavevmode
841   \PackageInfo{textcomp}{Faking \noexpand#1 for font family
842     \f@family\MessageBreak in TS1 encoding}%
843   \valign{##\cr
844     \vfil\hbox to 0.07em{\dimen@\f@size\p@
845       \math@fontsfalse
846       \fontsize{.7\dimen@}\z@\selectfont=\hss}%
847     \vfil\cr%
848     \hbox{C}\crcr
849   }%
850 }

```

\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.

```

851 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
852 \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. . .

```

853 \DeclareTextAccentDefault{\capitalcedilla}{TS1}
854 \DeclareTextAccentDefault{\capitalogonek}{TS1}
855 \DeclareTextAccentDefault{\capitalgrave}{TS1}
856 \DeclareTextAccentDefault{\capitalacute}{TS1}
857 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}
858 \DeclareTextAccentDefault{\capitaltilde}{TS1}
859 \DeclareTextAccentDefault{\capitaldieresis}{TS1}
860 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
861 \DeclareTextAccentDefault{\capitalring}{TS1}
862 \DeclareTextAccentDefault{\capitalcaron}{TS1}
863 \DeclareTextAccentDefault{\capitalbreve}{TS1}
864 \DeclareTextAccentDefault{\capitalmacron}{TS1}
865 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}

```

... and then the other glyphs.

```

866 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
867 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
868 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
869 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
870 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}
871 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
872 \DeclareTextSymbolDefault{\textdollar}{TS1}
873 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
874 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
875 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
876 \DeclareTextSymbolDefault{\textminus}{TS1}
877 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
878 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
879 \DeclareTextSymbolDefault{\textasciigrave}{TS1}

```

```

880 \DeclareTextSymbolDefault{\texttildelow}{TS1}
881 \DeclareTextSymbolDefault{\textasciibreve}{TS1}
882 \DeclareTextSymbolDefault{\textasciicaron}{TS1}
883 \DeclareTextSymbolDefault{\textgravedbl}{TS1}
884 \DeclareTextSymbolDefault{\textacutedbl}{TS1}
885 \DeclareTextSymbolDefault{\textdagger}{TS1}
886 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
887 \DeclareTextSymbolDefault{\textbardbl}{TS1}
888 \DeclareTextSymbolDefault{\textperthousand}{TS1}
889 \DeclareTextSymbolDefault{\textbullet}{TS1}
890 \DeclareTextSymbolDefault{\textcelsius}{TS1}
891 \DeclareTextSymbolDefault{\textflorin}{TS1}
892 \DeclareTextSymbolDefault{\texttrademark}{TS1}
893 \DeclareTextSymbolDefault{\textcent}{TS1}
894 \DeclareTextSymbolDefault{\textsterling}{TS1}
895 \DeclareTextSymbolDefault{\textyen}{TS1}
896 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
897 \DeclareTextSymbolDefault{\textsection}{TS1}
898 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}
899 \DeclareTextSymbolDefault{\textcopyright}{TS1}
900 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
901 \DeclareTextSymbolDefault{\textlnot}{TS1}
902 \DeclareTextSymbolDefault{\textregistered}{TS1}
903 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
904 \DeclareTextSymbolDefault{\textdegree}{TS1}
905 \DeclareTextSymbolDefault{\textpm}{TS1}
906 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
907 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
908 \DeclareTextSymbolDefault{\textasciiacute}{TS1}
909 \DeclareTextSymbolDefault{\textmu}{TS1}
910 \DeclareTextSymbolDefault{\textparagraph}{TS1}
911 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
912 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
913 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
914 \DeclareTextSymbolDefault{\textonequarter}{TS1}
915 \DeclareTextSymbolDefault{\textonehalf}{TS1}
916 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
917 \DeclareTextSymbolDefault{\texttimes}{TS1}
918 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

919 \DeclareTextCommandDefault{\texteuro}
920   {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```

921 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}

```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```

922 \DeclareTextCommandDefault{\textestimated}%
923   {\tc@check@symbol3\textestimated}
924 \DeclareTextCommandDefault{\textcurrency}%
925   {\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.

```

926 \DeclareTextCommandDefault{\capitaltie}%
927   {\tc@check@accent2\capitaltie}
928 \DeclareTextCommandDefault{\newtie}%
929   {\tc@check@accent2\newtie}
930 \DeclareTextCommandDefault{\capitalnewtie}%
931   {\tc@check@accent2\capitalnewtie}
932 \DeclareTextCommandDefault{\textleftarrow}%
933   {\tc@check@symbol2\textleftarrow}
934 \DeclareTextCommandDefault{\textrightarrow}%
935   {\tc@check@symbol2\textrightarrow}
936 \DeclareTextCommandDefault{\textblank}%
937   {\tc@check@symbol2\textblank}
938 \DeclareTextCommandDefault{\textdblhyphen}%
939   {\tc@check@symbol2\textdblhyphen}
940 \DeclareTextCommandDefault{\textzerooldstyle}%
941   {\tc@check@symbol2\textzerooldstyle}
942 \DeclareTextCommandDefault{\textoneoldstyle}%
943   {\tc@check@symbol2\textoneoldstyle}
944 \DeclareTextCommandDefault{\texttwooldstyle}%
945   {\tc@check@symbol2\texttwooldstyle}
946 \DeclareTextCommandDefault{\textthreeoldstyle}%
947   {\tc@check@symbol2\textthreeoldstyle}
948 \DeclareTextCommandDefault{\textfouroldstyle}%
949   {\tc@check@symbol2\textfouroldstyle}
950 \DeclareTextCommandDefault{\textfiveoldstyle}%
951   {\tc@check@symbol2\textfiveoldstyle}
952 \DeclareTextCommandDefault{\textsixoldstyle}%
953   {\tc@check@symbol2\textsixoldstyle}
954 \DeclareTextCommandDefault{\textsevenoldstyle}%
955   {\tc@check@symbol2\textsevenoldstyle}
956 \DeclareTextCommandDefault{\texteightoldstyle}%
957   {\tc@check@symbol2\texteightoldstyle}
958 \DeclareTextCommandDefault{\textnineoldstyle}%
959   {\tc@check@symbol2\textnineoldstyle}
960 \DeclareTextCommandDefault{\textlangle}%
961   {\tc@check@symbol2\textlangle}
962 \DeclareTextCommandDefault{\textrangle}%
963   {\tc@check@symbol2\textrangle}
964 \DeclareTextCommandDefault{\textmho}%
965   {\tc@check@symbol2\textmho}
966 \DeclareTextCommandDefault{\textbigcircle}%
967   {\tc@check@symbol2\textbigcircle}
968 \DeclareTextCommandDefault{\textuparrow}%
969   {\tc@check@symbol2\textuparrow}
970 \DeclareTextCommandDefault{\textdownarrow}%
971   {\tc@check@symbol2\textdownarrow}
972 \DeclareTextCommandDefault{\textborn}%
973   {\tc@check@symbol2\textborn}
974 \DeclareTextCommandDefault{\textdivorced}%
975   {\tc@check@symbol2\textdivorced}
976 \DeclareTextCommandDefault{\textdied}%
977   {\tc@check@symbol2\textdied}

```

```

978 \DeclareTextCommandDefault{\textleaf}%
979   {\tc@check@symbol2\textleaf}
980 \DeclareTextCommandDefault{\textmarried}%
981   {\tc@check@symbol2\textmarried}
982 \DeclareTextCommandDefault{\textmusicalnote}%
983   {\tc@check@symbol2\textmusicalnote}
984 \DeclareTextCommandDefault{\textdblhyphenchar}%
985   {\tc@check@symbol2\textdblhyphenchar}
986 \DeclareTextCommandDefault{\textdollaroldstyle}%
987   {\tc@check@symbol2\textdollaroldstyle}
988 \DeclareTextCommandDefault{\textcentoldstyle}%
989   {\tc@check@symbol2\textcentoldstyle}
990 \DeclareTextCommandDefault{\textcolonmonetary}%
991   {\tc@check@symbol2\textcolonmonetary}
992 \DeclareTextCommandDefault{\textwon}%
993   {\tc@check@symbol2\textwon}
994 \DeclareTextCommandDefault{\textnaira}%
995   {\tc@check@symbol2\textnaira}
996 \DeclareTextCommandDefault{\textguarani}%
997   {\tc@check@symbol2\textguarani}
998 \DeclareTextCommandDefault{\textpeso}%
999   {\tc@check@symbol2\textpeso}
1000 \DeclareTextCommandDefault{\textlira}%
1001   {\tc@check@symbol2\textlira}
1002 \DeclareTextCommandDefault{\textrecipe}%
1003   {\tc@check@symbol2\textrecipe}
1004 \DeclareTextCommandDefault{\textinterrobang}%
1005   {\tc@check@symbol2\textinterrobang}
1006 \DeclareTextCommandDefault{\textinterrobangdown}%
1007   {\tc@check@symbol2\textinterrobangdown}
1008 \DeclareTextCommandDefault{\textdong}%
1009   {\tc@check@symbol2\textdong}
1010 \DeclareTextCommandDefault{\textpertenthousand}%
1011   {\tc@check@symbol2\textpertenthousand}
1012 \DeclareTextCommandDefault{\textpilcrow}%
1013   {\tc@check@symbol2\textpilcrow}
1014 \DeclareTextCommandDefault{\textbaht}%
1015   {\tc@check@symbol2\textbaht}
1016 \DeclareTextCommandDefault{\textnumero}%
1017   {\tc@check@symbol2\textnumero}
1018 \DeclareTextCommandDefault{\textdiscount}%
1019   {\tc@check@symbol2\textdiscount}
1020 \DeclareTextCommandDefault{\textopenbullet}%
1021   {\tc@check@symbol2\textopenbullet}
1022 \DeclareTextCommandDefault{\textservicemark}%
1023   {\tc@check@symbol2\textservicemark}
1024 \DeclareTextCommandDefault{\textlquill}%
1025   {\tc@check@symbol2\textlquill}
1026 \DeclareTextCommandDefault{\textrquill}%
1027   {\tc@check@symbol2\textrquill}
1028 \DeclareTextCommandDefault{\textcopyleft}%
1029   {\tc@check@symbol2\textcopyleft}
1030 \DeclareTextCommandDefault{\textcircledP}%
1031   {\tc@check@symbol2\textcircledP}

```

```

1032 \DeclareTextCommandDefault{\textreferencemark}%
1033     {\tc@check@symbol2\textreferencemark}
1034 \DeclareTextCommandDefault{\textsurd}%
1035     {\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.

```

1036 \DeclareTextCommandDefault{\textcircled}
1037     {\CheckEncodingSubset\UseTextAccent{TS1}%
1038     {\UseTextAccent{OMS}}1\textcircled}
1039 \DeclareTextCommandDefault{\t}
1040     {\CheckEncodingSubset\UseTextAccent{TS1}%
1041     {\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).

```

1042 \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^AT_EX which will prevent the usage of the TS1 versions. So we better get rid of them:

```

1043 \UndeclareTextCommand{\textsterling}{OT1}
1044 \UndeclareTextCommand{\textdollar} {OT1}

```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```

1045 %\UndeclareTextCommand{\textsterling}{OT4}
1046 %\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`.

```

1047 \UndeclareTextCommand{\textperthousand}{T1}
1048 %\UndeclareTextCommand{\textpertenthousand}{T1}

```

61.1.1 Supporting oldstyle digits

```

1049 \DeclareRobustCommand\oldstylenums[1]{%
1050     \begingroup
1051     \ifmmode
1052         \mathgroup\symletters #1%

```

```

1053 \else
1054 \CheckEncodingSubset\@use@text@encoding{TS1}%
1055 {\PackageWarning{textcomp}%
1056 {Oldstyle digits unavailable for
1057 family \f@family.\MessageBreak
1058 Lining digits used instead}}%
1059 \tw@{#1}%
1060 \fi
1061 \endgroup
1062 }

```

61.1.2 Subset encoding defaults

For many font families commonly used in the T_EX 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:

```

1063 \iftc@forced \else

```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```

1064 \DeclareEncodingSubset{TS1}{cmr}      {0}
1065 \DeclareEncodingSubset{TS1}{cmss}     {0}
1066 \DeclareEncodingSubset{TS1}{cmtt}     {0}
1067 \DeclareEncodingSubset{TS1}{cmvtt}    {0}
1068 \DeclareEncodingSubset{TS1}{cmbr}     {0}
1069 \DeclareEncodingSubset{TS1}{cmtl}     {0}
1070 \DeclareEncodingSubset{TS1}{ccr}      {0}

```

PSNFSS fonts:

```

1071 \DeclareEncodingSubset{TS1}{ptm}      {4}
1072 \DeclareEncodingSubset{TS1}{pcr}      {4}
1073 \DeclareEncodingSubset{TS1}{phv}      {4}
1074 \DeclareEncodingSubset{TS1}{ppl}      {3}
1075 \DeclareEncodingSubset{TS1}{pag}      {4}
1076 \DeclareEncodingSubset{TS1}{pbk}      {4}
1077 \DeclareEncodingSubset{TS1}{pnc}      {4}
1078 \DeclareEncodingSubset{TS1}{pzc}      {4}
1079 \DeclareEncodingSubset{TS1}{bch}      {4}
1080 \DeclareEncodingSubset{TS1}{put}      {5}

```

Other CTAN fonts (probably not complete):

```

1081 \DeclareEncodingSubset{TS1}{uag}      {5}
1082 \DeclareEncodingSubset{TS1}{ugq}      {5}
1083 \DeclareEncodingSubset{TS1}{ul8}      {4}
1084 \DeclareEncodingSubset{TS1}{ul9}      {4} % (LuxiSans, one day)
1085 \DeclareEncodingSubset{TS1}{augie}    {5}
1086 \DeclareEncodingSubset{TS1}{dayrom}   {3}
1087 \DeclareEncodingSubset{TS1}{dayroms}  {3}
1088 \DeclareEncodingSubset{TS1}{pxr}      {0}
1089 \DeclareEncodingSubset{TS1}{pxss}     {0}
1090 \DeclareEncodingSubset{TS1}{pxtt}     {0}
1091 \DeclareEncodingSubset{TS1}{txr}      {0}
1092 \DeclareEncodingSubset{TS1}{txss}     {0}
1093 \DeclareEncodingSubset{TS1}{txtt}     {0}

```

Latin Modern and TeX Gyre:

```

1094 \DeclareEncodingSubset{TS1}{lmr}      {0}
1095 \DeclareEncodingSubset{TS1}{lmdh}      {0}
1096 \DeclareEncodingSubset{TS1}{lmss}      {0}
1097 \DeclareEncodingSubset{TS1}{lmssq}     {0}
1098 \DeclareEncodingSubset{TS1}{lmvtt}     {0}
1099 \DeclareEncodingSubset{TS1}{lmtt}      {0}

1100 \DeclareEncodingSubset{TS1}{qhv}       {0}
1101 \DeclareEncodingSubset{TS1}{qag}       {0}
1102 \DeclareEncodingSubset{TS1}{qbk}       {0}
1103 \DeclareEncodingSubset{TS1}{qcr}       {0}
1104 \DeclareEncodingSubset{TS1}{qcs}       {0}
1105 \DeclareEncodingSubset{TS1}{qpl}       {0}
1106 \DeclareEncodingSubset{TS1}{qtm}       {0}
1107 \DeclareEncodingSubset{TS1}{qzc}       {0}
1108 \DeclareEncodingSubset{TS1}{qhvc}      {0}

```

Fourier-GUTenberg:

```

1109 \DeclareEncodingSubset{TS1}{futs}      {4}
1110 \DeclareEncodingSubset{TS1}{futex}     {4}
1111 \DeclareEncodingSubset{TS1}{futj}       {4}

```

Y&Y's Lucida Bright

```

1112 \DeclareEncodingSubset{TS1}{hlh}       {3}
1113 \DeclareEncodingSubset{TS1}{hls}       {3}
1114 \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`.

```

1115 \DeclareEncodingSubset{TS1}{hlct}      {5}
1116 \DeclareEncodingSubset{TS1}{hlx}       {5}
1117 \DeclareEncodingSubset{TS1}{hlce}      {5}
1118 \DeclareEncodingSubset{TS1}{hlcn}      {5}
1119 \DeclareEncodingSubset{TS1}{hlcw}      {5}
1120 \DeclareEncodingSubset{TS1}{hlcf}      {5}

```

Other commercial families...

```

1121 \DeclareEncodingSubset{TS1}{pplx}      {3}
1122 \DeclareEncodingSubset{TS1}{pplj}      {3}
1123 \DeclareEncodingSubset{TS1}{ptmx}      {4}
1124 \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.

```

1125 \InputIfFileExists{textcomp.cfg}
1126 {\PackageInfo{textcomp}{Local configuration file used}}{}
1127 \fi
1128 \end{TS1oldsty}

```

File y

ltpageno.dtx

62 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 z

ltxref.dtx

63 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.

63.1 Cross Referencing

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

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

End of historical L^AT_EX 2.09 comments.

`\labelformat` A reference via `\ref` produces by default the data associated with the corresponding `\label` command (typically a number); any additional formatting has to be provided by the user. If, for example, references to equations are always to be typeset as “equation (*number*)”, one has to code “`equation (\ref{key})`”. With `\labelformat` there is a possibility to generate such frills automatically without resorting to low-level coding. The command takes two arguments: the first is the name of a counter and the second is its representation when referenced. This means that for a successful usage, one has to know the counter name being used for generating the label, though in practice this should not pose a problem. The current counter number is picked up as an argument. Here are two examples:

```
\labelformat{section}{section~#1}
\labelformat{equation}{equation~(#1)}
```

`\Ref` A side effect of using `\labelformat` is that, depending on the defined formatting, it becomes impossible to use `\ref` at the beginning of a sentence (if its replacement text starts with a lowercase letter). To overcome this problem we introduce the command `\Ref` that behave like `\ref` except that it uppercases the first token of the generated string.

To make `\Ref` work properly the very first token in the second argument of `\labelformat` has to be a simple ASCII or UTF-8 letter, otherwise the capitalization will fail or worse, you will end up with some error messages. If you actually need something more complicated in this place (e.g., an accented letter not written as a UTF-8 character) you have to explicitly surround it with braces, to identify the part that needs to be capitalized. For example, for figure references in the Hungarian language you might want to write `\labelformat{figure}{\a'bra~\thefigure}` or use `\labelformat{figure}{ábra~\thefigure}` which avoids the brace problem.

`\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.

`\@refundefined`

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_EX 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 redefined 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 once. It is executed by the `\enddocument` command.

```

31 \let \@multiplelabels \relax
```

`\label` The commands `\label` and `\refstepcounter` have been changed to allow `\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 \</2ekernel>
37 \<2ekernel | latexrelease>
38 \<latexrelease>\IncludeInRelease{2019/10/01}%
39 \<latexrelease>           {\refstepcounter}{Add \labelformat and \Ref}%

```

`\refstepcounter` Step the counter and allow for labels to point to its current value.

```
40 \def\refstepcounter#1{\stepcounter{#1}%  
41   \protected@edef\@currentlabel
```

By generating the second `csname` first the `\p@...` command can grab it as an argument which can be helpful for more complicated typesetting arrangements.

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 original 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.

```
42     {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%  
43 }
```

`\labelformat` A shortcut to set the `\p@...` macro for a counter. It will pick up the counter representation as an argument so that it can be specially formatted.

```
44 \def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

`\Ref` This macro expands the result of `\ref` and then uppercases the first token. Only useful if the label was generated via `\labelformat` and contains some lower case letter at its start. If the label starts with a complicated construct (e.g., an accented letter that is provided via a command, e.g., `\"a` instead of a UTF-8 character like ä) one has to surround everything that needs uppercasing in a brace group in the definition of `\labelformat`.¹²

```
45 \DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%  
46   \expandafter\MakeUppercase\@tempa}  
  
47 \if2ekernel\latexrelease  
48 \if\latexrelease\EndIncludeInRelease  
49 \if\latexrelease\IncludeInRelease{0000/00/00}%  
50 \if\latexrelease      {\refstepcounter}{Add \labelformat and \Ref}%  
51 \if\latexrelease  
52 \if\latexrelease\def\refstepcounter#1{\stepcounter{#1}%  
53 \if\latexrelease    \protected@edef\@currentlabel  
54 \if\latexrelease      {\csname p@#1\endcsname\csname the#1\endcsname}%  
55 \if\latexrelease}  
56 \if\latexrelease\let\labelformat\undefined  
57 \if\latexrelease\let\Ref\undefined  
58 \if\latexrelease  
59 \if\latexrelease\EndIncludeInRelease  
60 \if*2ekernel
```

`\@currentlabel` Default for `\label` commands that come before any environment.

```
61 \def\@currentlabel{}  
  
62 \if2ekernel
```

¹²There is one problem with this approach: the braces are kept in a normal `\ref` which might spoil kerning. Perhaps one day this needs redoing.

File A

ltmiscen.dtx

64 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 (*2kernel)
2 \message{environments,}
```

64.1 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}82$.

`\enddocument` is the user’s command for ending the manuscript file.

`\stop` is a panic button — to end $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ in the middle.

Historical $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X} 2.09$ comments (not necessarily accurate any more):

```
\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
      \bibtex{LABEL}{VAL} == null
      BEGIN
        \reserved@a == VAL
        if def(\reserved@a) = def(\g@LABEL)
```

```

else @tempswa := true      fi
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
\endgroup
finish up
END

\@writefile{EXT}{ENTRY} ==
if tf@EXT undefined
else \write\tf@EXT{ENTRY}
fi
End of historical LATEX 2.09 comments.

\@currenvir The name of the current environment.  Initialized to document to so that
\end{document} works correctly.
3 \def\@currenvir{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 \@tempswatrue \fi}
```

Reading data from auxiliary files (like `.toc` normally happens in vertical mode and it therefore doesn't matter if line endings are converted to spaces by `TeX` during that process.

However, especially the `.toc` file might be read in L-R mode (in cases the `\tableofcontents` attempts to put, say a list of sub-sections as a paragraph. In that case the newlines after a line like

```
\contentsline {subsubsection}{\numberline {1.1.1}A C-head}{2}
```

might result in spurious spaces (e.g., when that level is not included).

That could be fixed by reading in the file using `\endlinechar=-1` but that has the danger that it drops some valid endlines that should be converted to spaces (for example when the user edited the TOC and then used `\nofiles` to preserve it.

So the approach taken instead is this:

- `\addcontentsline` adds the command `\protected@file@percent` to the end of the second argument of `\@writefile` that is written to the `.aux`. As the name indicates this is a protected macro so it doesn't change if it is written out.
- When the `.aux` is read back in at the end of the run, `\@writefile` is executed and writes its second argument unmodified to the file with the extension given by its first argument. Or rather that was how it was in the past.
- Instead we change `\@writefile` slightly: basically it looks at the second argument and if the last token in there is `\protected@file@percent` then it is replaced by a percent character and that is then written out. If not (for example, if the data came from a user issued `\addtocontents`, or from some package that uses `\@writefile` for writing its own files) then the command behaves exactly as before.

`\protected@file@percent` Dummy cs to be replaced by a percent sign inside `\@writefile`. If it survives (when used incorrectly) it will expand to nothing in a typesetting context.

```
43 \<2kernel>
44 (*2kernel | latexrelease)
45 (latexrelease)\IncludeInRelease{2018/12/01}%
46 (latexrelease)          {\protected@file@percent}{Mask line endings}%
47 \protected\def\protected@file@percent{}
```

`\add@percent@to@temptokena` Helper function which is used to inspect a sequence of tokens (the second argument of `\@writefile` and if the last token is `\protected@file@percent` it will replace it by a harmless percent. The result is saved in `\@temptokena` for later use.

```
48 \catcode'\^^A=9
49 \long\gdef\add@percent@to@temptokena
50   #1\protected@file@percent#2\add@percent@to@temptokena
```

When we call this macro in `\@writefile` we stick in `\@empty` at the beginning, so that in case the tokenlist consists of a single brace group the braces aren't stripped. The `\expandafter` then expands this extra token away again.

```
51   {\expandafter\ifx\expandafter X\detokenize{#2}X\expandafter\dont@add@percent@to@temptokena
52     \expandafter\do@add@percent@to@temptokena\fi{#1}}
```

```

53 \long\def\dont@add@percent@to@temptokena#1{%
54   \@temptokena\expandafter{#1}}

```

`latexrelease` will read this code in high-speed mode in certain situations. During that it will only look for `\if` tests but not actually execute the `\catcode` change above. As a result it will drop anything after the `%` character in the definition. Therefore the `\fi` needs to be on the next line and we need locally another comment character to avoid getting spaces into the definition—a weird problem :-)

```

55 \begingroup
56 \catcode'\%=12
57 \catcode'\^^A=14
58 \long\gdef\do@add@percent@to@temptokena#1{\@temptokena\expandafter{#1%\^^A

```

Can't be on the same line as the `%` — see above.

```

59   }}
60 \endgroup

```

`\@writefile`

```

61 \long\def\@writefile#1#2{%
62   \@ifundefined{tf@#1}\relax
63   {%

```

If we write to the file we first prepare `#2` using `\add@percent@to@temptokena` and then write the token register out.

```

64     \add@percent@to@temptokena
65     \@empty#2\protected@file@percent
66     \add@percent@to@temptokena
67     \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
68   }%
69 }

70 \if2ekernel | latexrelease)
71 \if2ekernel \EndIncludeInRelease
72 \if2ekernel \IncludeInRelease{0000/00/00}%
73 \if2ekernel \protected@file@percent}{Mask line endings}%
74 \if2ekernel \let\protected@file@percent\@undefined
75 \if2ekernel \let\add@percent@to@temptokena\@undefined
76 \if2ekernel \let\do@add@percent@to@temptokena\@undefined
77 \if2ekernel \let\dont@add@percent@to@temptokena\@undefined
78 \if2ekernel \long\def\@writefile#1#2{%
79 \if2ekernel \ifundefined{tf@#1}\relax
80 \if2ekernel \ifundefined{tf@#1}\relax
81 \if2ekernel \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
82 \if2ekernel }%
83 \if2ekernel }
84 \if2ekernel \EndIncludeInRelease
85 \if2ekernel

```

`\stop`

```

86 \def\stop{\clearpage\deadcycles\z@\let\par\@par\@end}

```

Historical \LaTeX 2.09 comments (not necessarily accurate any more):

```

87 \everypar{\@nodocument} %% To get an error if text appears before the

```

```

88 \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

End of historical LATEX 2.09 comments.

\begin
89 </2kernel>
90 <*2kernel | latexrelease>
91 <latexrelease>\IncludeInRelease{2019/10/01}%
92 <latexrelease>                {\begin}{Making \begin/\end robust}%
93 \DeclareRobustCommand\begin[1]{%

```



```

94 \ifundefined{#1}%
95   {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
96   {\def\reserved@a{\def\@currentvir{#1}%
97     \edef\@currentvline{\on@line}%
98     \csname #1\endcsname}}%
99 \ignorefalse
100 \begingroup\@endpfalse\reserved@a}

```

A version that doesn't start out with `\relax` when in typesetting mode would be the following, but since `\begin` issues a `\begingroup` it wouldn't help much with respect to allowing things like `\noalign` or `\multicolumn` inside.

```

101 %\edef\begin
102 %  {\unexpanded{%
103 %    \ifx\protect\@typeset@protect
104 %      \expandafter\@gobble
105 %    \fi
106 %    \protect
107 %  }}%
108 %  \expandafter\noexpand\csname begin \endcsname
109 %  }
110 %\@namedef{begin }#1{%
111 %  \ifundefined{#1}%
112 %    {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
113 %    {\def\reserved@a{\def\@currentvir{#1}%
114 %      \edef\@currentvline{\on@line}%
115 %      \csname #1\endcsname}}%
116 %  \ignorefalse
117 %  \begingroup\@endpfalse\reserved@a}

```

`\end` While `\begin` was made robust simply by using `\DeclareRobustCommand` we need to be a bit more subtle with `\end` as there are packages out there that try to look into the top-level contents of `\end{foo}` (that is at the expansion of `\endfoo`) to see if it contains certain macros. This is done by hitting `\end{foo}` with three `\expandafters`, the first to get

```
\csname endfoo\endcsname \checkend{foo}% etc.
```

the second to expand the `\csname`, i.e., to get to

```
\endfoo \checkend{foo}% etc.
```

and the third to finally get to the top-level content of `\endfoo`, i.e.

```
<top-level content of \endfoo> \checkend{foo}% etc.
```

Therefore a robust replacement should produce the same results after three expansions (there first is obviously different).

Basically the definition of `\end` should either produce `\protect\end␣` (when not doing typesetting) or it should produce `\end␣` (without the `\protect`) when doing typesetting. Furthermore, it should (when in typesetting mode) show exactly the same result as `\end␣` (which is the original fragile definition of `\end`) when you expand either of them twice, i.e.,

```
\endfoo \checkend{foo}% etc.
```

That is achieved with the code below (which is worth studying carefully).

There is some trickery involved here: in particular we use `\romannumeral` to change a single expansion into three successive expansions in one go. That primitive expands until it has scanned a number (0 in this case, so it doesn't produce any output) and so it allows us to place arbitrary many `\expandafters` inside that are all going to be executed when `\romannumeral` is hit by a single `\expandafter`.

```

118 \edef\end
119   {\unexpanded{%
120     \romannumeral
121     \ifx\protect\@typeset@protect
122       \expandafter      %1
123       \expandafter      %2
124       \expandafter      %1
125       \expandafter      %3 expands the \csname inside \end<space>
126       \expandafter      %1
127       \expandafter      %2 expands \end<space>
128       \expandafter      %1 expands the \else
129       \z@
130     \else
131       \expandafter\z@\expandafter\protect
132     \fi
133   }%
134   \expandafter\noexpand\csname end \endcsname
135 }

```

And here is the original definition of `\end` the way it was in \LaTeX for several decades now hidden in `\end_`.

```

136 \@namedef{end }#1{%
137   \csname end#1\endcsname\@checkend{#1}%
138   \expandafter\endgroup\if@endpe\@doendpe\fi
139   \if@ignore\@ignorefalse\ignorespaces\fi}
140 </2ekernel | latexrelease>
141 <latexrelease>\EndIncludeInRelease

```

And here the rollback in case that is ever needed.

```

142 <latexrelease>\IncludeInRelease{0000/00/00}%
143 <latexrelease>          {\begin}{Making \begin/\end robust}%
144 <latexrelease>\kernel@make@fragile\begin
145 <latexrelease>\kernel@make@fragile\end
146 <latexrelease>
147 <latexrelease>\EndIncludeInRelease
148 <*2ekernel>

```

`\@checkend`

```

149 \def\@checkend#1{\def\reserved@a{#1}\ifx
150   \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.

```

151 \let\@currenvline\@empty

```

64.2 Center, Flushright, Flushleft

152 \message{center,}

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```
\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

End of historical L^AT_EX 2.09 comments.

\@centercr

```
153 \</2kernel>
154 \<*2kernel | latexrelease>
155 \<latexrelease>\IncludeInRelease{2020/02/02}%
156 \<latexrelease>          {\@centercr}{Make robust}%
157 \protected\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
158       \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
159 \</2kernel | latexrelease>

160 \<latexrelease>\EndIncludeInRelease
161 \<latexrelease>\IncludeInRelease{0000/00/00}%
162 \<latexrelease>          {\@centercr}{Make robust}%
163 \<latexrelease>
164 \<latexrelease>\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
165 \<latexrelease>          \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
166 \<latexrelease>
167 \<latexrelease>\EndIncludeInRelease
168 \<*2kernel>
```

\@xcentercr

```
169 \def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
170     [\@icentercr\ignorespaces}
```

\@icentercr

```
171 \def\@icentercr[#1]{\vskip #1\ignorespaces}
```

```

center We use \relax to prevent \item scanning too far.
172 \def\center{\trivlist \centering\item\relax}
173 \def\endcenter{\endtrivlist}

174 \if2kernel
175 \if*2kernel | latexrelease
176 \if latexrelease \IncludeInRelease{2019/10/01}%
177 \if latexrelease {\centering}{Make commands robust}%

\centering
178 \DeclareRobustCommand\centering{%
179   \let\\\@centercr
180   \rightskip\@flushglue\leftskip\@flushglue
181   \parindent\z@\parfillskip\z@skip}

\raggedright
182 \DeclareRobustCommand\raggedright{%
183   \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
184   \leftskip\z@skip
185   \parindent\z@}

\raggedleft
186 \DeclareRobustCommand\raggedleft{%
187   \let\\\@centercr
188   \rightskip\z@skip\leftskip\@flushglue
189   \parindent\z@\parfillskip\z@skip}

190 \if2kernel | latexrelease
191 \if latexrelease \EndIncludeInRelease
192 \if latexrelease \IncludeInRelease{0000/00/00}%
193 \if latexrelease {\centering}{Make commands robust}%
194 \if latexrelease
195 \if latexrelease \kernel@make@fragile\centering
196 \if latexrelease \kernel@make@fragile\raggedright
197 \if latexrelease \kernel@make@fragile\raggedleft
198 \if latexrelease
199 \if latexrelease \EndIncludeInRelease
200 \if*2kernel

\@rightskip
201 \newskip\@rightskip \@rightskip \z@skip

flushleft We use \relax to prevent \item scanning too far.
202 \def\flushleft{\trivlist \raggedright\item\relax}
203 \def\endflushleft{\endtrivlist}

flushright We use \relax to prevent \item scanning too far.
204 \def\flushright{\trivlist \raggedleft\item\relax}
205 \def\endflushright{\endtrivlist}

```

64.3 Verbatim

206 \message{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 TeXbook's space character instead of as blank spaces.

\@vobeyspaces

207 {\catcode'\ =\active%

208 \gdef\@vobeyspaces{\catcode'\ \active\let \@xobeysp}}

\@xobeysp

\@xverbatim

\@sxverbatim

```
209 \begingroup \catcode '|=0 \catcode '[= 1
210 \catcode']=2 \catcode '\{=12 \catcode '\}=12
211 \catcode'\|=12 \gdef\@xverbatim#1\end{verbatim} [#1\end[verbatim]]
212 \gdef\@sxverbatim#1\end{verbatim*} [#1\end[verbatim*]]
213 \endgroup
```

\@verbatim Real start of verbatim environment We use `\relax` to prevent `\item` scanning too far.

```
214 \</2ekernel>
215 \*2ekernel| latexrelease>
216 \<latexrelease>\IncludeInRelease{2017-04-15}{\@verbatim}%
217 \<latexrelease> {Disable hyphenation in verbatim}%
218 \def\@verbatim{\trivlist \item\relax
219 \if@minipage\else\vskip\parskip\fi
220 \leftskip\@totalleftmargin\rightskip\z@skip
221 \parindent\z@\parfillskip\@flushglue\parskip\z@skip
```

Added `\@@par` to clear possible `\parshape` definition from a surrounding list (the verbatim guru says). Switch language when in vertical mode.

222 \@@par

Set `\language` here to suppress hyphenation. Done this way rather than setting `\hyphenchar` as that is a global setting.

```
223 \language\l@nohyphenation
224 \@tempswafalse
225 \def\par{%
226 \if@tempswa
```

A `\leavevmode` added: needed if, for example, a blank verbatim line is the first thing in a list item (wow!).

```
227 \leavevmode \null \@@par\penalty\interlinepenalty
228 \else
229 \@tempswatrue
230 \ifhmode\@@par\penalty\interlinepenalty\fi
231 \fi}%
```

To allow customization we hide the font used in a separate macro.

```

232 \let\do\@makeother \dospecials
233 \obeylines \verbatim@font \@noligs

To avoid a breakpoint after the labels box, we remove the penalty put there by
the list macros: another use of \unpenalty!

234 \everypar \expandafter{\the\everypar \unpenalty}%
235 }
236 /2kernel | latexrelease>
237 (latexrelease)\EndIncludeInRelease
238 (latexrelease)\IncludeInRelease{0000-00-00}{\@verbatim}%
239 (latexrelease) {Disable hyphenation in verbatim}%
240 (latexrelease)\def\@verbatim{\trivlist \item\relax
241 (latexrelease) \if@minipage\else\vskip\parskip\fi
242 (latexrelease) \leftskip\@totalleftmargin\rightskip\z@skip
243 (latexrelease) \parindent\z@\parfillskip\@flushglue\parskip\z@skip
244 (latexrelease) @@par
245 (latexrelease) \@tempswafalse
246 (latexrelease) \def\par{%
247 (latexrelease) \if@tempswa
248 (latexrelease) \leavevmode \null @@par\penalty\interlinepenalty
249 (latexrelease) \else
250 (latexrelease) \@tempswatrue
251 (latexrelease) \ifhmode@@par\penalty\interlinepenalty\fi
252 (latexrelease) \fi}%
253 (latexrelease) \let\do\@makeother \dospecials
254 (latexrelease) \obeylines \verbatim@font \@noligs
255 (latexrelease) \hyphenchar\font\m@ne
256 (latexrelease) \everypar \expandafter{\the\everypar \unpenalty}%
257 (latexrelease)}
258 (latexrelease)\EndIncludeInRelease
259 (*2kernel)

```

\verbatim (RmS 93/09/19) Protected against ‘missing item’ error message triggered by empty verbatim environment.

```

260 \def\verbatim{\@verbatim \frenchspacing\@vobeyspaces \@xverbatim}
261 \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.

```

262 \def\verbatim@font{\normalfont\ttfamily}

263 /2kernel)
264 (*2kernel | latexrelease>
265 (latexrelease)\IncludeInRelease{2018/12/01}%
266 (latexrelease) {\verbvisiblespace}{Setup visible space for verb}%

```

\asciispace The character in slot 32, in typewriter fonts (historically) a visible space but in other fonts a real space or something else

```

267 \DeclareRobustCommand\asciispace{\char 32 }

```

\verbvisiblespace This defines how to get a visible space in **\verb*** and friends. In classic T_EX this is just the slot 32, but in TU encoded fonts we switch fonts and take the character from cmtt.

```

268 \ifx\Umathcode\@undefined
269   \let\verbvisiblespace\asciispace % Pdftex version
270 \else
271   \DeclareRobustCommand\verbvisiblespace
272     {\leavevmode{\usefont{OT1}{cmtt}{m}{n}\asciispace}} % xetex/luatex version
273 \fi

```

`\@setupverbvisiblespace` In pdf \TeX a catcode 12 space will produce the character in slot 32 which is assumed to be a visible space character (in a typewriter font in OT1 or T1 encoding). In Xe \TeX or Lua \TeX a font in TU encoding is normally used and that has a real space in this slot. So what we do in this case is this: we check the definition of `\verbvisiblespace` and if it is `\asciispace` we assume that the char32 can be used (e.g., in pdf \TeX). We then redefine `\@xobeysp` so that after running `\@vobeyspaces` we get characters from slot 32 for each active space.

```

274 \def\@setupverbvisiblespace{%
275   \ifx\verbvisiblespace\asciispace
276     \let\@xobeysp\asciispace
277   \else

```

Otherwise we measure the width of a character in the mon-spaced current font and place a `\verbvisiblespace` into a box of the right width which we are then using as the character for a space. By default this will be the space character from OT1 cmtt but by changing `\verbvisiblespace` one could use, for example, the `\textvisiblespace` of the current typewriter font.

```

278     \setbox\z@\hbox{x}%
279     \setbox\@verbvisiblespacebox\hbox to\wd\z@{\hss\verbvisiblespace\hss}%
280     \def\@xobeysp{\leavevmode\copy\@verbvisiblespacebox}%
281   \fi
282 }

```

`\@verbvisiblespacebox` The box to hold the visible space character if it isn't in slot 32 in the current typewriter font.

```

283 \newbox\@verbvisiblespacebox

```

`\@sverb` Definitions of `\@sverb` and `\@verb` changed so `\verb+foo+` does not lose leading blanks when it comes at the beginning of a line. Change made 24 May 89. Suggested by Frank Mittelbach and Rainer Schöpf.

```

284 \def\@sverb#1{%
285   \catcode'\#1\active
286   \lccode'\~'#1%
287   \gdef\verb@balance@group{\verb@egroup
288     \@latexerror{\noexpand\verb illegal in command argument}\@ehc}%
289   \aftergroup\verb@balance@group
290   \lowercase{\let~\verb@egroup}%

```

If `\@sverb` is called from `\@verb` then space is already active and supposed to produce a real space. In this case we do nothing. Otherwise we run `\@setupverbvisiblespace` to setup the right visible space char and afterwards `\@vobeyspaces` to make it the definition for the active space character.

```

291   \ifnum\catcode'\ =\active
292     \else \@setupverbvisiblespace \@vobeyspaces \fi
293 }

```

```

verbatim* For verbatim* we also set up the correct visible space character definition and
then run \@vobeyspaces. As this code is not called as part of the normal verbatim
environment (the method is done the other way around this time) we don't have
to check if space is already active—it shouldn't be.

294 \@namedef{verbatim*}{\@verbatim
295   \@setupverbvisiblespace
296   \frenchspacing\@vobeyspaces\@sxverbatim}
297 \expandafter\let\csname endverbatim*\endcsname=\endverbatim

298 </2ekernel | latexrelease>
299 (latexrelease)\EndIncludeInRelease
300 (latexrelease)\IncludeInRelease{0000/00/00}%
301 (latexrelease)           {\verbvisiblespace}{Setup visible space for verb}%
302 (latexrelease)
303 (latexrelease)\@namedef{verbatim*}{\@verbatim\@sxverbatim}
304 (latexrelease)
305 (latexrelease)\let\asciispace           \@undefined
306 (latexrelease)\let\verbvisiblespace     \@undefined
307 (latexrelease)\let\@setupverbvisiblespace\@undefined
308 (latexrelease)\let\@verbvisiblespacebox \@undefined
309 (latexrelease)
310 (latexrelease)\def\@sverb#1{%
311 (latexrelease)  \catcode'#1\active
312 (latexrelease)  \lccode'\~'#1%
313 (latexrelease)  \gdef\verb@balance@group{\verb@egroup
314 (latexrelease)    \@latex@error{noexpand\verb illegal in command argument}\@ehc}%
315 (latexrelease)  \aftergroup\verb@balance@group
316 (latexrelease)  \lowercase{\let~\verb@egroup}}%
317 (latexrelease)
318 (latexrelease)\EndIncludeInRelease
319 (*2ekernel)

\@makeoother

320 \def\@makeoother#1{\catcode'#112\relax}

\verb@balance@group

321 \let\verb@balance@group\@empty

\verb@egroup

322 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}

\verb@eol@error

323 \begingroup
324   \obeylines%
325   \gdef\verb@eol@error{\obeylines%
326     \def~M{\verb@egroup\@latex@error{%
327       \noexpand\verb ended by end of line}\@ehc}}%
328 \endgroup

\verb Typesetting a small piece verbatim.

329 </2ekernel>
330 (*2ekernel | latexrelease)
331 (latexrelease)\IncludeInRelease{2017-04-15}{\verb}%

```



```

332 \latexrelease) {Disable hyphenation in verb}%
333 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
334 \bgroup
335 \verb@eol@error \let\do\@makeother \dospecials
336 \verbatim@font\@noligs

```

Set \language here to suppress hyphenation. Done this way rather than setting \hyphenchar as that is a global setting.

```

337 \language\l@nohyphenation
338 \ifstar\@sverb\@verb}
339 \endkernel\latexrelease)
340 \latexrelease)\EndIncludeInRelease
341 \latexrelease)\IncludeInRelease{0000-00-00}{\verb}%
342 \latexrelease) {Disable hyphenation in verb}%
343 \latexrelease)\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
344 \latexrelease) \bgroup
345 \latexrelease) \verb@eol@error \let\do\@makeother \dospecials
346 \latexrelease) \verbatim@font\@noligs
347 \latexrelease) \ifstar\@sverb\@verb}
348 \latexrelease)\EndIncludeInRelease
349 \endkernel)

```

\@verb

```

350 \def\@verb{\@vobeyspaces \frenchspacing \@sverb}

```

\verbatim@nolig@list

```

351 \def\verbatim@nolig@list{\do\'\do\<\do\>\do\,\do\'\do\~}

```

\do@noligs

```

352 \def\do@noligs#1{%
353 \catcode'#1\active
354 \begingroup
355 \lccode'\~'#1\relax
356 \lowercase{\endgroup\def~{\leavevmode\kern\z@\char'#1}}

```

\@noligs To stay compatible with packages that use \@noligs we keep it.

```

357 \def\@noligs{\let\do\do@noligs \verbatim@nolig@list}

```

```

358 \endkernel)

```

File B

ltmath.dtx

65 Math setup

This file contains a lot of the original plain T_EX code, as well as the L^AT_EX environments for math. It still needs sorting out.

```
1 (*2kernel)
2 \message{math definitions,}
```

65.1 Math commands based on plain T_EX

65.1.1 The log-like functions

\log The standard operators:

```
3 \DeclareRobustCommand\log{\mathop{\operator@font log}\nolimits}
4 \DeclareRobustCommand\lg{\mathop{\operator@font lg}\nolimits}
5 \DeclareRobustCommand\ln{\mathop{\operator@font ln}\nolimits}
6 \DeclareRobustCommand\lim{\mathop{\operator@font lim}}
7 \DeclareRobustCommand\limsup{\mathop{\operator@font lim}\,sup}}
8 \DeclareRobustCommand\liminf{\mathop{\operator@font lim}\,inf}}
9 \DeclareRobustCommand\sin{\mathop{\operator@font sin}\nolimits}
10 \DeclareRobustCommand\arcsin{\mathop{\operator@font arcsin}\nolimits}
11 \DeclareRobustCommand\sinh{\mathop{\operator@font sinh}\nolimits}
12 \DeclareRobustCommand\cos{\mathop{\operator@font cos}\nolimits}
13 \DeclareRobustCommand\arccos{\mathop{\operator@font arccos}\nolimits}
14 \DeclareRobustCommand\cosh{\mathop{\operator@font cosh}\nolimits}
15 \DeclareRobustCommand\tan{\mathop{\operator@font tan}\nolimits}
16 \DeclareRobustCommand\arctan{\mathop{\operator@font arctan}\nolimits}
17 \DeclareRobustCommand\tanh{\mathop{\operator@font tanh}\nolimits}
18 \DeclareRobustCommand\cot{\mathop{\operator@font cot}\nolimits}
19 \DeclareRobustCommand\coth{\mathop{\operator@font coth}\nolimits}
20 \DeclareRobustCommand\sec{\mathop{\operator@font sec}\nolimits}
21 \DeclareRobustCommand\csc{\mathop{\operator@font csc}\nolimits}
22 \DeclareRobustCommand\max{\mathop{\operator@font max}}
23 \DeclareRobustCommand\min{\mathop{\operator@font min}}
24 \DeclareRobustCommand\sup{\mathop{\operator@font sup}}
25 \DeclareRobustCommand\inf{\mathop{\operator@font inf}}
26 \DeclareRobustCommand\arg{\mathop{\operator@font arg}\nolimits}
27 \DeclareRobustCommand\ker{\mathop{\operator@font ker}\nolimits}
28 \DeclareRobustCommand\dim{\mathop{\operator@font dim}\nolimits}
29 \DeclareRobustCommand\hom{\mathop{\operator@font hom}\nolimits}
30 \DeclareRobustCommand\det{\mathop{\operator@font det}}
31 \DeclareRobustCommand\exp{\mathop{\operator@font exp}\nolimits}
32 \DeclareRobustCommand\Pr{\mathop{\operator@font Pr}}
33 \DeclareRobustCommand\gcd{\mathop{\operator@font gcd}}
34 \DeclareRobustCommand\deg{\mathop{\operator@font deg}\nolimits}
```

\bmod And some operators have to be done by hand:

```
35 \DeclareRobustCommand\bmod{%
36   \nonscript\mskip-\medmuskip\mkern5mu%
```

```

37 \mathbin{\operator@font mod}\penalty900\mkern5mu%
38 \nonscript\mskip-\medmuskip}

\pmod
39 \DeclareRobustCommand\pmod[1]{%
40 \allowbreak\mkern18mu({\operator@font mod}\,\,\,#1)}

```

65.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```

41 \DeclareRobustCommand\bigl{\mathopen\big}
42 \DeclareRobustCommand\bigr{\mathrel\big}
43 \DeclareRobustCommand\bigr{\mathclose\big}
44 \DeclareRobustCommand\Bigl{\mathopen\Big}
45 \DeclareRobustCommand\Bigr{\mathrel\Big}
46 \DeclareRobustCommand\Bigr{\mathclose\Big}
47 \DeclareRobustCommand\biggl{\mathopen\bigg}
48 \DeclareRobustCommand\biggm{\mathrel\bigg}
49 \DeclareRobustCommand\biggr{\mathclose\bigg}
50 \DeclareRobustCommand\Biggl{\mathopen\Bigg}
51 \DeclareRobustCommand\Biggm{\mathrel\Bigg}
52 \DeclareRobustCommand\Biggr{\mathclose\Bigg}

```

65.1.3 The UNSORTED Rest

The other math commands are lifted from plain T_EX.

```

\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 \</2ekernel>
78 \<*2ekernel | latexrelease>
79 \<latexrelease>\IncludeInRelease{2019/10/01}%
80 \<latexrelease> \vphantom{Make commands robust}%
81 \DeclareRobustCommand\vphantom{\v@true\h@false\ph@nt}
82 \DeclareRobustCommand\hphantom{\v@false\h@true\ph@nt}
83 \DeclareRobustCommand\phantom{\v@true\h@true\ph@nt}

\mathstrut

84 \DeclareRobustCommand\mathstrut{\vphantom{}}

85 \</2ekernel | latexrelease>
86 \<latexrelease>\EndIncludeInRelease
87 \<latexrelease>\IncludeInRelease{0000/00/00}%
88 \<latexrelease> \vphantom{Make commands robust}%
89 \<latexrelease>
90 \<latexrelease>\kernel@make@fragile\vphantom
91 \<latexrelease>\kernel@make@fragile\hphantom
92 \<latexrelease>\kernel@make@fragile\phantom
93 \<latexrelease>\kernel@make@fragile\mathstrut
94 \<latexrelease>
95 \<latexrelease>\EndIncludeInRelease
96 \<*2ekernel>

97 \def\ph@nt{%
98 \ifmmode
99 \expandafter\mathpalette\expandafter\mathph@nt
100 \else
101 \expandafter\makeph@nt
102 \fi}

103 \def\makeph@nt#1{%
104 \setbox\z@ \hbox{\color@begingroup#1\color@endgroup}\finph@nt}

105 \def\mathph@nt#1#2{%
106 \setbox\z@ \hbox{\$ \m@th#1{#2}$}\finph@nt}

```

```

107 </2ekernel>
108 <*2ekernel | latexrelease>
109 <latexrelease>\IncludeInRelease{2018/12/01}%
110 <latexrelease>          {\finph@nt}{Start LR-mode}%
111 \def\finph@nt{%
112   \setbox\tw@ \null
113   \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
114   \ifh@ \wd\tw@\wd\z@\fi

115   \leavevmode@ifvmode\box\tw@}
116 </2ekernel | latexrelease>
117 <latexrelease>\EndIncludeInRelease
118 <latexrelease>\IncludeInRelease{0000/00/00}%
119 <latexrelease>          {\finph@nt}{Start LR-mode}%
120 <latexrelease>\def\finph@nt{%
121 <latexrelease>   \setbox\tw@ \null
122 <latexrelease>   \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
123 <latexrelease>   \ifh@ \wd\tw@\wd\z@\fi \box\tw@}
124 <latexrelease>\EndIncludeInRelease
125 <*2ekernel>

\smash

126 \DeclareRobustCommand\smash{%
127   \relax % \relax, in case this comes first in \halign
128   \ifmmode
129     \expandafter\mathpalette\expandafter\mathsm@sh
130   \else
131     \expandafter\makesm@sh
132   \fi}

133 \def\makesm@sh#1{%
134   \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
135 \def\mathsm@sh#1#2{%
136   \setbox\z@\hbox{\$m@th#1{#2}$}\finsm@sh}

137 </2ekernel>
138 <*2ekernel | latexrelease>
139 <latexrelease>\IncludeInRelease{2018/12/01}%
140 <latexrelease>          {\finsm@sh}{Start LR-mode}%
141 \def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \leavevmode@ifvmode\box\z@}
142 </2ekernel | latexrelease>
143 <latexrelease>\EndIncludeInRelease
144 <latexrelease>\IncludeInRelease{0000/00/00}%
145 <latexrelease>          {\finsm@sh}{Start LR-mode}%
146 <latexrelease>\def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}
147 <latexrelease>\EndIncludeInRelease
148 <*2ekernel>

\buildrel

149 \def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}}

150 </2ekernel>
151 <*2ekernel | latexrelease>
152 <latexrelease>\IncludeInRelease{2019/10/01}%
153 <latexrelease>          {\cases}{Make commands robust}%

```

```

\cases
154 \DeclareRobustCommand\cases[1]{\left\{\,\,\vcenter{\normalbaselines\m@th
155 \ialign{##$\hfil$\&\quad{##}\hfil\crr#1\crr}\}\right.}

\matrix
156 \DeclareRobustCommand\matrix[1]{\null\,\vcenter{\normalbaselines\m@th
157 \ialign{\hfil$##$\hfil\&\quad\hfil$##$\hfil\crr
158 \mathstrut\crr\noalign{\kern-\baselineskip}
159 #1\crr\mathstrut\crr\noalign{\kern-\baselineskip}}}\,,}

\pmatrix
160 \DeclareRobustCommand\pmatrix[1]{\left(\matrix{#1}\right)}

161 (/2kernel | latexrelease)
162 (latexrelease)\EndIncludeInRelease
163 (latexrelease)\IncludeInRelease{0000/00/00}%
164 (latexrelease) \cases{Make commands robust}%
165 (latexrelease)
166 (latexrelease)\kernel@make@fragile\cases
167 (latexrelease)\kernel@make@fragile\matrix
168 (latexrelease)\kernel@make@fragile\pmatrix
169 (latexrelease)
170 (latexrelease)\EndIncludeInRelease
171 (*2kernel)

\bordermatrix
172 \def\bordermatrix#1{\begingroup \m@th
173 \@tempdima 8.75\p@
174 \setbox\z@\vbox{%
175 \def\cr{\crr\noalign{\kern2\p@\global\let\cr\endline}}%
176 \ialign{##$\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
177 &\quad\hfil$##$\hfil\crr
178 \omit\strut\hfil\crr\noalign{\kern-\baselineskip}%
179 #1\crr\omit\strut\cr}}%
180 \setbox\tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
181 \setbox\tw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
182 \setbox\tw@\hbox{$\kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
183 \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
184 \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,,\right)$}%
185 \null\;\vbox{\kern\ht\@ne\box\tw@}\endgroup}

\openup
186 \def\openup{\afterassignment\openup\dimen@}
187 \def\@openup{\advance\lineskip\dimen@
188 \advance\baselineskip\dimen@
189 \advance\lineskiplimit\dimen@}

\displaylines
190 \newif\ifdt@p
191 \def\display{\global\dt@ptrue\openup\jot\m@th
192 \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
193 \vskip-\lineskiplimit \vskip\normallineskiplimit \fi
194 \else \penalty\interdisplaylinepenalty \fi}}}

```

```

195 \def\@align{\tabskip\z@skip\everycr{}} % restore inside \displ@y
196 \def\displaylines#1{\displ@y \tabskip\z@skip
197   \halign{\hb@xt@\displaywidth{${\@align\hfil\displaystyle##\hfil$}\crrc
198     #1\crrc}}

\sp
\sb 199 \let\sp=^
    200 \let\sb=_

\>
\; 201 %\def\,{\mskip\thinmuskip}      % already defined in ltspac
\! 202 \def\>{\mskip\medmuskip}
    203 \def\;{\mskip\thickmuskip}
    204 \def\!{\mskip-\thinmuskip}

\*
205 \DeclareRobustCommand\*{\discretionary{\thinspace\the\textfont2\char2}{-}{-}}

\: Nickname for the medium space since \> is not available inside tabbing.
206 \let\:=\>

\active@math@prime This is the definition of the active math prime.
207 \def\active@math@prime{^{\bgroup\prim@s}}

\prime@s
208 {\catcode'\=' \active \global\let'\active@math@prime}
209 \def\prim@s{%
210   \prime\futurelet\@let@token\pr@m@s}
211 \def\pr@m@s{%
212   \ifx'\@let@token
213     \expandafter\pr@@@s
214   \else
215     \ifx^\@let@token
216       \expandafter\expandafter\expandafter\pr@@@t
217     \else
218       \egroup
219     \fi
220   \fi}
221 \def\pr@@@s#1{\prim@s}
222 \def\pr@@@t#1#2{#2\egroup}

223 {\catcode'\_=\active \gdef\_{} } % _ in math is
224                                     % either subscript or \_

```

65.2 Math Environments

`\(` Produces $\$. \dots \$$ with checks that `\(` isn't used in math mode, and that `\)` is only used in math mode begun with `\(`.

```

225 \endkernel
226 \ifx\latexrelease\IncludeInRelease{2015/01/01}\{\}\{Make \(\ robust}\%
227 \ifx\latexrelease\endkernel\endkernel\endkernel
228 \DeclareRobustCommand\(\{%
229   \relax\ifmmode\@badmath\else$\fi}%
230 \DeclareRobustCommand\)\{%
231   \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
232 \endkernel\ifx\latexrelease\endkernel\endkernel
233 \ifx\latexrelease\IncludeInRelease{0000/00/00}\{\}\{Make \(\ robust}\%
234 \ifx\latexrelease\endkernel\endkernel
235 \def\(\{%
236   \relax\ifmmode\@badmath\else$\fi}%
237 \ifx\latexrelease\endkernel\endkernel\let\csname\string\ \endcsname\@undefined
238 \def\)\{%
239   \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
240 \ifx\latexrelease\endkernel\endkernel\let\csname\string\ \endcsname\@undefined
241 \ifx\latexrelease\endkernel\endkernel
242 \endkernel

```

`\[` Produces $$$ \dots $$$ with checks that `\[` isn't used in math mode, and that `\]` is only used in display math mode (though there is no real test that this display math started with `\[` and not with $$$$).

```

243 \endkernel
244 \ifx\latexrelease\IncludeInRelease{2015/01/01}\{\}\{Make \[ robust}\%
245 \ifx\latexrelease\endkernel\endkernel
246 \DeclareRobustCommand\[{\%
247   \relax\ifmmode
248     \@badmath
249   \else
250     \ifvmode
251       \nointerlineskip
252       \makebox[.6\linewidth]{}%
253     \fi
254     $$$ BRACE MATCH HACK
255   \fi
256 }%
257 \DeclareRobustCommand\]\{%
258   \relax\ifmmode
259     \ifinner
260       \@badmath
261     \else
262       $$$ BRACE MATCH HACK
263     \fi
264   \else
265     \@badmath
266   \fi
267   \ignorespaces
268 }%
269 \endkernel\ifx\latexrelease\endkernel\endkernel

```



```

270 \latexrelease\EndIncludeInRelease
271 \latexrelease\IncludeInRelease{0000/00/00}\[{}]{Make \[ robust}%
272 \latexrelease\def\[{}%
273 \latexrelease\relax\ifmmode
274 \latexrelease\@badmath
275 \latexrelease\else
276 \latexrelease\ifvmode
277 \latexrelease\nointerlineskip
278 \latexrelease\makebox[.6\linewidth]{}%
279 \latexrelease\fi
280 \latexrelease$$$$ BRACE MATCH HACK
281 \latexrelease\fi
282 \latexrelease}%
283 \latexrelease\expandafter\let\csname\string[ \endcsname\@undefined
284 \latexrelease\def\[{}%
285 \latexrelease\relax\ifmmode
286 \latexrelease\ifinner
287 \latexrelease\@badmath
288 \latexrelease\else
289 \latexrelease$$$$ BRACE MATCH HACK
290 \latexrelease\fi
291 \latexrelease\else
292 \latexrelease\@badmath
293 \latexrelease\fi
294 \latexrelease\ignorespaces
295 \latexrelease}%
296 \latexrelease\expandafter\let\csname\string[ \endcsname\@undefined
297 \latexrelease\EndIncludeInRelease
298 \*2ekernel)

math Disguises for \(...\.) and \[...\].
displaymath 299 \let\math=\(
300 \let\endmath=\)
301 \def\displaymath{[{}
302 \def\enddisplaymath{\}\@ignoretrue}

equation \c@equation Numbered equations, using the counter \c@equation. Note: The document style
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.)
303 \definecounter{equation}
304 \def\equation{$$\refstepcounter{equation}}
305 \def\endequation{\eqno \hbox{\@eqnnum}$$\@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.
306 \def\@eqnnum{\normalfont \normalcolor (\theequation)}

\stackrel A disguise for plain TEX's buildrel.
307 \DeclareRobustCommand\stackrel[2]{\mathrel{\mathop{#2}\limits^{#1}}}
```

`\frac` A disguise for plain T_EX's `\over`.
308 `\DeclareRobustCommand\frac[2]{\begingroup#1\endgroup\over#2}`

`\sqrt` Add an optional argument to plain's `\sqrt` to give the n th root of an expression
`\@sqrt` $\sqrt[n]{e}$.
309 `\DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}`
310 `\def\@sqrt[#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\@eqnrel = \hfil`.
`\@eqpen` 311 `\newcount\@eqcnt`
`\if@eqnsw` 312 `\newcount\@eqpen`
`\@eqnrel` 313 `\newif\if@eqnsw\@eqnswtrue`
314 `\newskip\@centering`
315 `\@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.

```

316 \def\eqnarray{%
317   \stepcounter{equation}%
318   \def\@currentlabel{\p@equation\theequation}%
319   \global\@eqnswtrue
320   \m@th
321   \global\@eqcnt\z@
322   \tabskip\@centering
323   \let\\\@eqnrel
324   $$\everycr{}\halign to\displaywidth\bgroup
325     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnrel
326     &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil${##}$\hfil
327     &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
328     $\displaystyle{##}$\hfil\tabskip\@centering
329     &\global\@eqcnt\thr@@ \hbxt@\z@\bgroup\hss##\egroup
330     \tabskip\z@skip
331     \cr
332 }
333 \def\endeqnarray{%
334   \@eqnrel
335   \egroup
336   \global\advance\c@equation\m@ne
337   $$\@ignoretrue
338 }
339 \let\@eqnrel=\relax

```

`\nonumber` Switches off equation numbering.
340 `\def\nonumber{\global\@eqnswfalse}`

`\@eqnrel`
`\@xeqnrel` 341 `\def\@eqnrel{%`
`\@yeqnrel` 342 `{\ifnum0='}\fi`
343 `\@ifstar{%`
344 `\global\@eqpen\@M\@yeqnrel`

```

345 }{%
346 \global\@eqpen\interdisplaylinepenalty \@eqnocr
347 }%
348 }

349 \def\@yeqnocr{\@testopt\@xeqnocr\z@skip}

350 \def\@xeqnocr[#1]{%
351 \ifnum0='{ \fi}%
352 \@eqnocr
353 \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
354 }

\@eqnocr
355 \def\@eqnocr{\let\reserved@a\relax
356 \ifcase\@eqcnt \def\reserved@a{& &}\or \def\reserved@a{& &}%
357 \or \def\reserved@a{&}\else
358 \let\reserved@a\empty
359 \latexerror{Too many columns in eqnarray environment}\@ehc\fi
360 \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
361 \global\@eqnswtrue\global\@eqcnt\z@\cr}

eqnarray* Here's the eqnarray* environment:
\@seqnocr 362 \let\@seqnocr=\@eqnocr
363 \@namedef{eqnarray*}{\def\@eqnocr{\nonumber\@seqnocr}\eqnarray}
364 \@namedef{endeqnarray*}{\nonumber\endeqnarray}

\lefteqn \lefteqn{FORMULA} typesets FORMULA in display math style flushleft in a box of
width zero.
365 \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.
366 \DeclareRobustCommand{\ensuremath}{%
367 \ifmmode
368 \expandafter\@firstofone
369 \else
370 \expandafter\@ensuredmath
371 \fi}

\@ensuredmath The \relax stops \ensuremath{} starting display math.
372 \long\def\@ensuredmath#1{\$ \relax #1$}
373 \endkernel

```

65.3 External options to the standard document classes

65.3.1 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.

```

374 (*leqno)
375 \renewcommand\@eqnnum{\hb@xt@.01\p@{}}%
376 \rlap{\normalfont\normalcolor
377 \hskip -\displaywidth(\theequation)}}
378 \leqno

```

65.3.2 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^AT_EX 2_ε's displayed math environments.

```

\mathindent The amount of indentation of the equations is stored in a register.
379 (*fleqn)
380 \newdimen\mathindent

The setting of \mathindent has to be deferred until the class file has been pro-
cessed, because \leftmargini is still 0pt wide at the moment fleqn.clo is read
in.
381 \AtEndOfClass{\mathindent\leftmargini}

\[ Begin display math;
382 \IncludeInRelease{2015/01/01}{\[\]{Make \[ robust}%
383 \DeclareRobustCommand\[\relax
384 \ifmmode\@badmath
385 \else
386 \begin{trivlist}%
387 \beginparpenalty\predisplaypenalty
388 \endparpenalty\postdisplaypenalty
389 \item[]\leavevmode
390 \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
391 \hskip\mathindent\bgroup
392 \fi}
393 \EndIncludeInRelease

394 \IncludeInRelease{0000/00/00}{\[\]{Make \[ robust}%
395 \renewcommand\[\relax
396 \ifmmode\@badmath
397 \else
398 \begin{trivlist}%
399 \beginparpenalty\predisplaypenalty
400 \endparpenalty\postdisplaypenalty
401 \item[]\leavevmode
402 \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
403 \hskip\mathindent\bgroup
404 \fi}
405 \EndIncludeInRelease

\] end display math;
406 \IncludeInRelease{2015/01/01}{\]\]{Make \] robust}%
407 \DeclareRobustCommand\]\relax
408 \ifmmode
409 \egroup $\hfil% $

```

```

410             \egroup
411             \end{trivlist}%
412             \else \@badmath
413             \fi}
414 \EndIncludeInRelease

415 \IncludeInRelease{0000/00/00}{\[]}{Make \] robust}%
416 \renewcommand\]{\relax
417             \ifmmode
418                 \egroup $\hfil% $
419                 \egroup
420                 \end{trivlist}%
421                 \else \@badmath
422                 \fi}
423 \EndIncludeInRelease

```

equation The equation environment

```

424 \renewenvironment{equation}%
425 {\@beginparpenalty\predisplaypenalty
426   \@endparpenalty\postdisplaypenalty
427   \refstepcounter{equation}%
428   \trivlist \item[]\leavevmode
429     \hb@xt@\linewidth\bgroup $\m@th% $
430     \displaystyle
431     \hskip\mathindent}%
432   {$\hfil % $
433     \displaywidth\linewidth\hbox{\@eqnnum}%
434   \egroup
435   \endtrivlist}

```

eqnarray The eqnarray environment

```

436 \renewenvironment{eqnarray}{%
437   \stepcounter{equation}%
438   \def\@currentlabel{\p@equation\theequation}%
439   \global\@eqnswtrue\m@th
440   \global\@eqcnt\z@
441   \tabskip\mathindent
442   \let\=\@eqnocr
443   \setlength\abovedisplayskip{\topsep}%
444   \ifvmode
445     \addtolength\abovedisplayskip{\partopsep}%
446   \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.

```

447   \addtolength\abovedisplayskip{\parskip}%
448   \setlength\belowdisplayskip{\abovedisplayskip}%
449   \setlength\belowdisplayshortskip{\abovedisplayskip}%
450   \setlength\abovedisplayshortskip{\abovedisplayskip}%
451   $$\everycr{}\halign to\linewidth% $$
452   \bgroup
453     \hskip\@centering
454     $\displaystyle\tabskip\z@skip{##}$\@eqnrel&%
455     \global\@eqcnt\@ne \hskip \tw@\arraycolsep \hfil${##}$\hfil&%

```

```

456     \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
457     $\displaystyle{##}$\hfil \tabskip\@centering&%
458     \global\@eqcnt\thr@@
459     \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}%
460     {\@eqnocr
461     \egroup
462     \global\advance\c@equation\m@ne$$% $$
463     \@ignoretrue
464     }
465 \</fleqn>

```

File C

ltlists.dtx

66 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.

66.1 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`.

`\@nparitem` A switch set by `\list` when `\@inlabel = true`. Handles the case of a `\list` being the first thing in an item.

`\@nparlist` 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^AT_EX 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.

66.2 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.

66.3 Penalties

`\@beginparpenalty`: put at the beginning of a list

`\@endparpenalty`: put at end of list

`\@itempenalty`: put between items.

66.4 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.

66.5 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. *Historical L^AT_EX 2.09 comments (not necessarily accurate any more):*

```
\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 @noperitem :=L true
      @noperlist :=L true
    else @noperlist :=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 @noperlist = true
      else if \lastskip > 0
        then \@tempskipa := \lastskip
        \vskip - \lastskip
        \vskip \@tempskipa - \@outerparskip + \parskip
      fi
    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
\begingroup %% 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
End of historical LATEX 2.09 comments.

\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 \@nmblistfalse
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_EX. 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@nolist \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-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@nobreak
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@}

```

66.6 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`.

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```
\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
```

End of historical L^AT_EX 2.09 comments.

```
\@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

Historical LATEX 2.09 comments (not necessarily accurate any more):
\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

End of historical LATEX 2.09 comments.

\@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 \</2ekernel>

```

File D

ltboxes.dtx

67 L^AT_EX 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: s stretched, l flushleft, r flushright, c (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 s, l, r or c (default) for stretched, flushleft, flushright or centred, and t or b for top, bottom – or combinations like tr or rb. Default for horizontal and vertical are centered. Note that in this picture mode version of <code>\makebox</code> a [b] 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 _E X box register.
<code>\savebox</code>	<code>\savebox{⟨cmd⟩} ...</code> : <code>⟨cmd⟩</code> is defined to be a T _E 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: c : <code>\vcenter</code> (placed in <code>\$...\$</code> if not in math mode) b : <code>\vbox</code> t : <code>\vtop</code> default value is c. 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 *2\kernel \rangle
2 \message{boxes,}

\makebox \makebox User level command just looks for optional [ or (.
3 \langle /2\kernel \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>\expandafter\let\csname makebox \endcsname\undefined
22 <latexrelease>\EndIncludeInRelease
23 <*2ekernel>

\mbox The basic horizontal box command for LATEX.
24 \DeclareRobustCommand\mbox[1]{\leavevmode\hbox{#1}}

\@makebox Look for a possible second optional argument (defaults to c).
25 \def\@makebox[#1]{%
26   \@ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}

\@begin@tempboxa Helper macro for supporting \height, \width etc. Grab #1 into \@tempboxa and
measure it.
27 \long\def\@begin@tempboxa#1#2{%
28   \begingroup
29   \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
30   \def\width{\wd\@tempboxa}%
31   \def\height{\ht\@tempboxa}%
32   \def\depth{\dp\@tempboxa}%
33   \let\totalheight\@ovri
34   \totalheight\height
35   \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.
36 \let\@end@tempboxa\endgroup

\bm@c Set up spacing.
\bm@l 37 \def\bm@c{\hss\unhbox\@tempboxa\hss}
\bm@r 38 \def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
\bm@s 39 \def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
\bm@t 40 \def\bm@s{\unhbox\@tempboxa}
\bm@b
\@imakebox Internal form of \makebox.
41 \long\def\@imakebox[#1][#2]#3{%
42   \@begin@tempboxa\hbox{#3}%
43   \setlength\@tempdima{#1}%          support calc
44   \hb@xt@\@tempdima{\csname bm@#2\endcsname}%
45   \@end@tempboxa}

```


`\@makepicbox` Picture mode form of `\makebox`.

```
46 \def\@makepicbox(#1,#2){%
47   \ifnextchar[{\@makepicbox(#1,#2)}{\@makepicbox(#1,#2) []}}
```

`\imakepicbox` picture mode version

```
48 \long\def\imakepicbox(#1,#2)[#3]#4{%
49   \vbox to#2\unitlength
50     {\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
51      \let\mb@t\vss
52      \@tfor\reserved@a :=#3\do{%
53        \if s\reserved@a
54          \let\mb@l\relax\let\mb@r\relax
55        \else
56          \expandafter\let\csname mb@\reserved@a\endcsname\relax
57        \fi}%
58      \mb@t
59      \hb@xt@ #1\unitlength{\mb@l #4\mb@r}%
60      \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.

```
61   \kern\z@}}
```

`\set@color` This macro is initially a no-op, but the colour package will redefine it to insert a `\special`.

```
62 \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`, `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and *(set to main document colour)* respectively.

```
\color@hbox 63 \let\color@begingroup\relax
\color@vbox 64 \let\color@endgroup\relax
\color@endbox 65 \let\color@setgroup\relax
66 \let\normalcolor\relax
67 \let\color@hbox\relax
68 \let\color@vbox\relax
69 \let\color@endbox\relax
```

`\newsavebox` Allocate a new ‘savebox’.

```
70 \def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}
```

`\savebox` Save `#1` in a box register.

```
71 \</2ekernel>
72 \<latexrelease>\IncludeInRelease{2015/01/01}%
73 \<latexrelease>      {\savebox}{Make \savebox robust}%
74 \<*2ekernel | latexrelease>
75 \DeclareRobustCommand\savebox[1]{%
76   \@ifnextchar(%
77     {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
78 \</2ekernel | latexrelease>
79 \<latexrelease>\EndIncludeInRelease
```

```

80 \latexrelease\IncludeInRelease{0000/00/00}%
81 \latexrelease\savebox\Make \savebox robust}%
82 \latexrelease\def\savebox#1{%
83 \latexrelease\ifnextchar(
84 \latexrelease\@savepicbox#1\@ifnextchar[\@savebox#1\@sbox#1}}%
85 \latexrelease\expandafter\let\csname savebox \endcsname\undefined
86 \latexrelease\EndIncludeInRelease
87 \*2kernel)

\savebox Save #1 in a box register.
88 \DeclareRobustCommand\savebox[2]{\setbox#1\hbox{%
89 \color@setgroup#2\color@endgroup}}

\@savebox Look for second optional argument.
90 \def\@savebox#1[#2]{%
91 \ifnextchar [\@isavebox#1[#2]]{\@isavebox#1[#2][c]}}

\@isavebox
92 \long\def\@isavebox#1[#2][#3]#4{%
93 \sbox#1{\@imakebox[#2][#3]{#4}}}

\@savepicbox Picture mode version of \savebox.
94 \def\@savepicbox#1(#2,#3){%
95 \ifnextchar[%
96 {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3)[]}}

\@isavepicbox Picture mode version of \savebox.
97 \long\def\@isavepicbox#1(#2,#3)[#4]#5{%
98 \sbox#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.
99 \def\lrbox#1{%
100 \edef\reserved@a{%
101 \endgroup
102 \setbox#1\hbox{%
103 \begingroup\aftergroup}%
104 \def\noexpand\@currentenv{\@currentenv}%
105 \def\noexpand\@currentline{\on@line}}%
106 \reserved@a
107 \@endpfalse
108 \color@setgroup
109 \ignorespaces}

\endlrbox End the lrbox environment.
110 \def\endlrbox{\unskip\color@endgroup}

\usebox unchanged
111 \DeclareRobustCommand\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`

```

112 \DeclareRobustCommand\frame[1]{%
113   \leavevmode
114   \hbox{%
115     \hskip-\@wholewidth
116     \vbox{%
117       \vskip-\@wholewidth
118       \hrule \@height\@wholewidth
119       \hbox{%
120         \vrule\@width\@wholewidth
121         #1%
122         \vrule\@width\@wholewidth}%
123       \hrule \@height\@wholewidth
124       \vskip-\@wholewidth}%
125     \hskip-\@wholewidth}}
```

`\fboxrule` user level parameters,

`\fboxsep` 126 `\newdimen\fboxrule`
127 `\newdimen\fboxsep`

`\fbox` Abbreviated framed box command.

```

128 \DeclareRobustCommand\fbox[1]{%
129   \leavevmode
130   \setbox\@tempboxa\hbox{%
131     \color@begingroup
132     \kern\fboxsep{#1}\kern\fboxsep
133     \color@endgroup}%
134   \@frameb@x\relax}
```

`\framebox` Framed version of `\makebox`.

```

135 \if2ekernel
136 \iflatexrelease\IncludeInRelease{2015/01/01}%
137 \iflatexrelease{\framebox}{Make \framebox robust}%
138 \if*2ekernel\iflatexrelease
139 \DeclareRobustCommand\framebox{%
140   \@ifnextchar(%)
141     \@framepicbox{\@ifnextchar[\@framebox\fbox}}%
142 \if2ekernel\iflatexrelease
143 \iflatexrelease\EndIncludeInRelease
144 \iflatexrelease\IncludeInRelease{0000/00/00}%
145 \iflatexrelease{\framebox}{Make \framebox robust}%
146 \iflatexrelease\def\framebox{%
147 \iflatexrelease   \@ifnextchar(%)
148 \iflatexrelease     \@framepicbox{\@ifnextchar[\@framebox\fbox}}%
149 \iflatexrelease\expandafter\let\csname framebox \endcsname\@undefined
150 \iflatexrelease\EndIncludeInRelease
151 \if*2ekernel
```

`\@framebox` Deal with optional arguments.

```

152 \def\@framebox[#1]{%
153   \@ifnextchar[%]
```

```

154     {\@iframebox[#1]}%
155     {\@iframebox[#1][c]}}

\@iframebox 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.
156 \long\def\@iframebox[#1][#2]#3{%
157   \leavevmode
158   \@begin@tempboxa\hbox{#3}%
159   \setlength\@tempdima{#1}%
160   \setbox\@tempboxa\hbext@\@tempdima
161     {\kern\fboxsep\csname bm@#2\endcsname\kern\fboxsep}%
162   \@frameb@x{\kern-\fboxrule}%
163   \@end@tempboxa}

\@frameb@x Common part of \framebox and \fbox. #1 is a negative kern in the \framebox
case so that the vertical rules do not add to the width of the box.
164 \def\@frameb@x#1{%
165   \@tempdima\fboxrule
166   \advance\@tempdima\fboxsep
167   \advance\@tempdima\dp\@tempboxa
168   \hbox{%
169     \lower\@tempdima\hbox{%
170       \vbox{%
171         \hrule\@height\fboxrule
172         \hbox{%
173           \vrule\@width\fboxrule
174           #1%
175           \vbox{%
176             \vskip\fboxsep
177             \box\@tempboxa
178             \vskip\fboxsep}%
179           #1%
180           \vrule\@width\fboxrule}%
181         \hrule\@height\fboxrule}%
182         }%
183       }%
184   }

\@framepicbox Picture mode version.
185 \def\@framepicbox(#1,#2){%
186   \@ifnextchar[{\@iframepicbox(#1,#2)}{\@iframepicbox(#1,#2) []}}

\@iframepicbox Picture mode version.
187 \long\def\@iframepicbox(#1,#2)[#3]#4{%
188   \frame{\@makepicbox(#1,#2)[#3]{#4}}

\parbox The main vertical-box command for LATEX.
189 </2ekernel>
190 <latexrelease>\IncludeInRelease{2015/01/01}%
191 <latexrelease>           {\parbox}{Make \parbox robust}%
192 <*2ekernel | latexrelease>
193 \DeclareRobustCommand\parbox{%

```

```

194 \@ifnextchar[%]
195 \@iparbox
196 {\@iiiparbox c\relax[s]}}%
197 /2ekernel | latexrelease)
198 (latexrelease)\EndIncludeInRelease
199 (latexrelease)\IncludeInRelease{0000/00/00}%
200 (latexrelease) {\parbox}{Make \parbox robust}%
201 (latexrelease)\def\parbox{%
202 (latexrelease) \@ifnextchar[%]
203 (latexrelease) \@iparbox
204 (latexrelease) {\@iiiparbox c\relax[s]}}%
205 (latexrelease)\expandafter\let\csname parbox \endcsname\@undefined
206 (latexrelease)\EndIncludeInRelease
207 (*2ekernel)

```

`\@iparbox` Optional argument handling.

```

208 \def\@iparbox[#1]{%
209 \@ifnextchar[%]
210 {\@iiiparbox{#1}}%
211 {\@iiiparbox{#1}\relax[s]}}

```

`\@iiiparbox` Optional argument handling.

```

212 \def\@iiiparbox#1[#2]{%
213 \@ifnextchar[%]
214 {\@iiiparbox{#1}{#2}}%
215 {\@iiiparbox{#1}{#2}[#1]}}

```

`\@iiiparbox` The internal version of `\parbox`.

```

\@parboxto
216 \let\@parboxto\@empty
217 \long\def\@iiiparbox#1#2[#3]#4#5{%
218 \leavevmode
219 \@pboxswfalse
220 \setlength\@tempdima{#4}%
221 \@begin@tempboxa\vbox{\hsize\@tempdima\@parboxrestore#5\@par}%
222 \ifx\relax#2\else
223 \setlength\@tempdimb{#2}%
224 \edef\@parboxto{to\the\@tempdimb}%
225 \fi
226 \if#1b\vbox
227 \else\if #1t\vtop
228 \else\ifmmode\vcenter
229 \else\@pboxswtrue $\vcenter
230 \fi\fi\fi
231 \@parboxto{\let\hss\vss\let\unhbox\unvbox
232 \csname bm#3\endcsname}%
233 \if@pboxsw \m@th$\fi
234 \@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@nbreak`; otherwise this command will be redundant.

```

235 \end{kernel}
236 \let\@arrayparboxrestore\@arrayparboxrestore
237 \let\@arrayparboxrestore\@arrayparboxrestore
238 \let\@arrayparboxrestore\@arrayparboxrestore
239 \let\@arrayparboxrestore\@arrayparboxrestore
240 \def\@arrayparboxrestore{%
241   \let\if@nbreak\iffalse
242   \let\if@noskipsec\iffalse
243   \let\par\@par
244   \let\-\@dischyph
245   \let\'\@acci\let\'\@accii\let\=\@acciii
246   \parindent\z@ \parskip\z@skip
247   \everypar{}%
248   \linewidth\hsize
249   \@totalleftmargin\z@
250   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
251   \parfillskip\@flushglue
252   \lineskip\normallineskip
253   \lineskiplimit\normallineskiplimit
254   \baselineskip\normalbaselineskip
255   \sloppy}
256 \end{kernel}
257 \let\@arrayparboxrestore\@arrayparboxrestore
258 \let\@arrayparboxrestore\@arrayparboxrestore
259 \let\@arrayparboxrestore\@arrayparboxrestore
260 \let\@arrayparboxrestore\@arrayparboxrestore
261 \def\@arrayparboxrestore{%
262   \let\if@nbreak\iffalse
263   \let\if@noskipsec\iffalse
264   \let\par\@par
265   \let\-\@dischyph
266   \let\'\@acci\let\'\@accii\let\=\@acciii
267   \parindent\z@ \parskip\z@skip
268   \everypar{}%
269   \linewidth\hsize
270   \@totalleftmargin\z@
271   \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
272   \parfillskip\@flushglue \lineskip\normallineskip
273   \baselineskip\normalbaselineskip
274   \sloppy}
275 \end{kernel}
276 \let\@arrayparboxrestore\@arrayparboxrestore

```

`\parboxrestore` Restore various paragraph parameters, and also `\.`

```

277 \def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}

```

```

\if@minipage Switch that is true at the start of a minipage.
278 \def\@minipagefalse{\global\let\if@minipage\iffalse}
279 \def\@minipagetrue {\global\let\if@minipage\iftrue}
280 \@minipagefalse

\minipage Essentially an environment form of \parbox.
281 \def\minipage{%
282   \@ifnextchar[%]
283     \@iminipage
284     {\@iiiminipage c\relax[s]}}

\@iminipage Optional argument handling.
285 \def\@iminipage[#1]{%
286   \@ifnextchar[%]
287     {\@iiiminipage{#1}}%
288     {\@iiiminipage{#1}\relax[s]}}

\@iiiminipage Optional argument handling.
289 \def\@iiiminipage#1[#2]{%
290   \@ifnextchar[%]
291     {\@iiiiminipage{#1}{#2}}%
292     {\@iiiiminipage{#1}{#2}[#1]}}

\@iiiiminipage Internal form of minipage.
293 \def\@iiiiminipage#1#2[#3]#4{%
294   \leavevmode
295   \@pboxswfalse
296   \setlength\@tempdima{#4}%
297   \def\@mpargs{{#1}{#2}[#3]{#4}}%
298   \setbox\@tempboxa\vbox\bgroup
299     \color@begingroup
300       \hsize\@tempdima
301       \textwidth\hsize \columnwidth\hsize
302       \@parboxrestore
303       \def\@mpfn{\mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
304       \let\@footnotetext\@mpfootnotetext
305       \let\@listdepth\@mplistdepth \@mplistdepth\z@
306       \@minipagerestore
307       \@setminipage}

\@minipagerestore Hook so that other styles can reset other commands in a minipage.
308 \let\@minipagerestore=\relax

\endminipage
309 \def\endminipage{%
310   \par
311   \unskip
312   \ifvoid\@mpfootins\else
313     \vskip\skip\@mpfootins
314     \normalcolor
315     \footnoterule
316     \unvbox\@mpfootins
317   \fi

```

```

318 \minipagefalse %% added 24 May 89
319 \color@endgroup
320 \egroup
321 \expandafter\@iiparbox\@mpargs{\unvbox\@tempboxa}}

\@mplistdepth Versions of \@listdepth and \footins local to minipage.
\@mpfootins 322 \newcount\@mplistdepth
323 \newinsert\@mpfootins

\@mpfootnotetext Minipage version of \@footnotetext.
Final \strut added 27 Mar 89, on suggestion by Don Hosek
324 \long\def\@mpfootnotetext#1{%
325 \global\setbox\@mpfootins\vbox{%
326 \unvbox\@mpfootins
327 \reset@font\footnotesize
328 \hsize\columnwidth
329 \@parboxrestore
330 \protected@edef\@currentlabel
331 {\csname p@mpfootnote\endcsname\@thefnmark}%
332 \color@begingroup
333 \@makefntext{%
334 \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
335 \color@endgroup}}

336 \newif\if@pboxsw

\@rule Draw a rule of the specified size.
337 (/2ekernel)
338 (latexrelease)\IncludeInRelease{2015/01/01}%
339 (latexrelease) {\rule}{Make \rule robust}%
340 (*2ekernel | latexrelease)
341 \DeclareRobustCommand\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
342 (/2ekernel | latexrelease)
343 (latexrelease)\EndIncludeInRelease
344 (latexrelease)\IncludeInRelease{0000/00/00}%
345 (latexrelease) {\rule}{Make \rule robust}%
346 (latexrelease)\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
347 (latexrelease)\expandafter\let\csname rule \endcsname\@undefined
348 (latexrelease)\EndIncludeInRelease
349 (*2ekernel)

\@rule Internal form of \rule.
350 \def\@rule[#1]#2#3{%
351 \leavevmode
352 \hbox{%
353 \setlength\@tempdima{#1}%
354 \setlength\@tempdimb{#2}%
355 \setlength\@tempdimc{#3}%
356 \advance\@tempdimc\@tempdima
357 \vrule\@width\@tempdimb\@height\@tempdimc\@depth-\@tempdima}}

\@@underline Saved primitive \underline.
358 \let\@@underline\underline

```


`\underline` L^AT_EX version works outside math.

```

359 \DeclareRobustCommand\underline[1]{%
360   \relax
361   \ifmmode\@@underline{#1}%
362   \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}

```

`\raisebox` Raise a box, and change its vertical dimensions.

```

363 \ifkernel
364 \ifkernel\IncludeInRelease{2015/01/01}%
365 \ifkernel\IncludeInRelease{2015/01/01}%
366 \ifkernel\IncludeInRelease{2015/01/01}%
367 \DeclareRobustCommand\raisebox[1]{%
368   \leavevmode
369   \ifnextchar[\@rsbox{#1}]{\@irsbox{#1}[]}}
370 \ifkernel\IncludeInRelease{2015/01/01}%
371 \ifkernel\IncludeInRelease{2015/01/01}%
372 \ifkernel\IncludeInRelease{2015/01/01}%
373 \ifkernel\IncludeInRelease{2015/01/01}%
374 \ifkernel\IncludeInRelease{2015/01/01}%
375 \ifkernel\IncludeInRelease{2015/01/01}%
376 \ifkernel\IncludeInRelease{2015/01/01}%
377 \ifkernel\IncludeInRelease{2015/01/01}%
378 \ifkernel\IncludeInRelease{2015/01/01}%
379 \ifkernel\IncludeInRelease{2015/01/01}%

```

`\@rsbox` Optional argument handling.

```

380 \def\@rsbox#1[#2]{%
381   \ifnextchar[\@iirsbox{#1}[#2]}{\@irsbox{#1}[#2]}}

```

`\@argsbox` ...

`\@irsbox` Internal version of `\raisebox` (less than two optional args).

```

382 \long\def\@irsbox#1[#2][#3]{%
383   \begin@tempboxa\hbox{#3}%
384   \setlength\@tempdima{#1}%
385   \ifx\#2\else\setlength\@tempdimb{#2}\fi
386   \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
387   \ifx\#2\else\ht\@tempboxa\@tempdimb\fi
388   \box\@tempboxa
389   \end@tempboxa}

```

`\@iirsbox` Internal version of `\raisebox` (two optional args).

```

390 \long\def\@iirsbox#1[#2][#3][#4]{%
391   \begin@tempboxa\hbox{#4}%
392   \setlength\@tempdima{#1}%
393   \setlength\@tempdimb{#2}%
394   \setlength\dimen@{#3}%
395   \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
396   \ht\@tempboxa\@tempdimb
397   \dp\@tempboxa\dimen@
398   \box\@tempboxa
399   \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.

```
400 \def\@finalstrut#1{%
401   \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}
```

67.1 Some low-level constructs

The following commands are basically inherited from plain T_EX.

<code>\leftline</code>	These macros place text on a full line either centred or left or right adjusted.
<code>\rightline</code>	
<code>\centerline</code>	
<code>\@@line</code>	

```

402 \def\@@line{\hb@xt@\hsize}
403 \DeclareRobustCommand\leftline[1]{\@@line{#1\hss}}
404 \DeclareRobustCommand\rightline[1]{\@@line{\hss#1}}
405 \DeclareRobustCommand\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.

406 \DeclareRobustCommand\rlap[1]{\hb@xt@\z@{#1\hss}}
407 \DeclareRobustCommand\llap[1]{\hb@xt@\z@{\hss#1}}

408 \endkernel
```

File E

lftab.dtx

68 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^AT_EX 2.09 version, not the extended version described in *The L^AT_EX Companion*. Use the `array` package to obtain the extended version.

68.1 tabbing

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

`\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 \@nxxtabmar < \@hightab
    then \@nxxtabmar :=G \@nxxtabmar+1
    else error message "Undefined tab"
    fi
END

\@tabminus ==
BEGIN
  if \@nxxtabmar > \@firsttab
    then \@nxxtabmar :=G \@nxxtabmar-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

```

End of historical L^AT_EX 2.09 comments.

- \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 \def\@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 \def\@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 \def\@startfield{%
47     \global\setbox\@curfield\hbox\bgroup\color@begingroup}

```



```

\@stopfield
48 \def\@stopfield{%
49   \color@endgroup\egroup}

\@contfield
50 \def\@contfield{%
51   \global\setbox\@curfield\hbox\bgroup\color@begingroup
52   \unhbox\@curfield}

\@addfield
53 \def\@addfield{\global\setbox\@curline\hbox{\unhbox
54   \@curline\unhbox\@curfield}}

\@ifatmargin
55 \def\@ifatmargin{\ifdim \wd\@curline =\z@}

\@tabcr
56 \def\@tabcr{\@stopline \ifstar{\penalty \@M \@xtabcr}\@xtabcr}

\@xtabcr
57 \def\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}

\@itabcr
58 \def\@itabcr[#1]{\vskip #1\@startline\ignorespaces}

\tabbing We use \relax to prevent \item from scanning too far.
59 \def\tabbing{\lineskip \z@skip\let\>\@rtab\let\<\@ltab\let\@=\@settab
60   \let\+\@tabplus\let\-\@tabminus\let\'\@tabrj\let\'\@tablab
61   \let\@=\@tabcr
62   \@hightab\@firsttab
63   \global\@nxttabmar\@firsttab
64   \dimen\@firsttab\@totalleftmargin
65   \global\@tabpush\z@ \global\@rjfieldfalse
66   \trivlist \item\relax
67   \if@minipage\else\vskip\parskip\fi

68   \setbox\@tabfbox\hbox{%
69     \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
70   \def\@itemfudge{\box\@tabfbox}%
71   \@startline\ignorespaces}

\endtabbing
72 \def\endtabbing{%
73   \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}

\@rtab Omitted \global added to \@rtab 17 Jun 86
74 \def\@rtab{\@stopfield\@addfield\ifnum \@curtab<\@hightab
75   \global\advance\@curtab \@ne \else\@badtab\fi
76   \@tempdima\dimen\@curtab
77   \advance\@tempdima -\dimen\@curtabmar
78   \advance\@tempdima -\wd\@curline
79   \global\setbox\@curline\hbox{\unhbox\@curline\hskip\@tempdima}%
80   \@startfield\ignorespaces}

```

\@settab

```
81 \def\@settab{\@stopfield\@addfield
82   \ifnum \@curtab <\@maxtab
83     \ifnum\@curtab =\@hightab
84       \advance\@hightab \@ne
85     \fi
86     \global\advance\@curtab \@ne
87   \else
88     \latexerror{Tab overflow}\@ehd
89   \fi
90   \dimen\@curtab \dimen\@curtabmar
91   \advance\dimen\@curtab \wd\@curline
92   \@startfield
93   \ignorespaces}
```

\@ltab

```
94 \def\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firsttab
95   \global\advance\@curtab \m@ne \global\advance\@curtabmar\m@ne\else
96   \@badtab\fi\else
97   \latexerror{string<\space in mid line}\@ehd\fi\ignorespaces}
```

\@tabplus

```
98 \def\@tabplus{%
99   \ifnum\@nxttabmar<\@hightab
100     \global\advance\@nxttabmar\@ne
101   \else
102     \@badtab
103   \fi
104   \ignorespaces}
```

\@tabminus

```
105 \def\@tabminus{%
106   \ifnum\@nxttabmar>\@firsttab
107     \global\advance\@nxttabmar\m@ne
108   \else
109     \@badtab
110   \fi
111   \ignorespaces}
```

\@tabrj

```
112 \def\@tabrj{%
113   \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}
```

\@tablab \setbox\@curline made \global in \@tablab. 17 Jun 86

```
114 \def\@tablab{%
115   \@stopfield
116   \global\setbox\@curline\hbox{%
117     \box\@curline
118     \hskip-\wd\@curfield \hskip-\tabbingsep
119     \box\@curfield
120     \hskip\tabbingsep}%
121   \@startfield
122   \ignorespaces}
```

```

123 </2ekernel>
124 <*2ekernel | latexrelease>
125 <latexrelease>\IncludeInRelease{2019/10/01}%
126 <latexrelease>                {\pushtabs}{Make commands robust}%

\pushtabs
127 \DeclareRobustCommand\pushtabs{%
128   \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
129     \@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.
130 \DeclareRobustCommand\poptabs{\@stopfield\@addfield
131   \ifnum \@tabpush >\z@
132     \endgroup
133     \global\advance\@tabpush \m@ne
134     \ifnum \@curtab >\@hightab
135       \global \@curtab \@hightab
136       \@badtab
137     \fi
138   \else
139     \@badpoptabs
140   \fi
141   \@contfield}

142 \DeclareRobustCommand\kill{\@stopfield\@startline\ignorespaces}

143 </2ekernel | latexrelease>
144 <latexrelease>\EndIncludeInRelease
145 <latexrelease>\IncludeInRelease{0000/00/00}%
146 <latexrelease>                {\pushtabs}{Make commands robust}%
147 <latexrelease>
148 <latexrelease>\kernel@make@fragile\pushtabs
149 <latexrelease>\kernel@make@fragile\poptabs
150 <latexrelease>\kernel@make@fragile\kill
151 <latexrelease>
152 <latexrelease>\EndIncludeInRelease
153 <*2ekernel>

\tabbingsep
154 \newdimen\tabbingsep

```

68.2 array and tabular environments

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

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
      \\\ == \@tabularcrrc
      \@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\crrc %%}
    \@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
    }
  END

```

```

\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 $
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
\cr
else \cr \noalign{\vskip LENGTH}
END

\@tabularcr and \@argtabularcr same as \@arraycr and
\@argarraycr
except without the extra $'s.
End of historical LATEX 2.09 comments.

\extracolsep This command needs to expand during the tabular preamble construction so can't
be robust.
155 \def\extracolsep#1{\tabskip #1\relax}

\array
156 \def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
157 \let\@classiv\@arrayclassiv
158 \let\\\@arraycr\let\@halignto\@empty\@tabarray}

\endarray
\endtabular 159 \def\endarray{\crcr\egroup\egroup}
\endtabular* 160 \def\endtabular{\crcr\egroup\egroup $\egroup}
161 \expandafter \let \csname endtabular*\endcsname = \endtabular

\tabular
162 \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.
163 \@namedef{tabular*}#1{%
164 \setlength\dimen@{#1}%
165 \edef\@halignto{to\the\dimen@}\@tabular}

```

```

\@tabular
166 \def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
167 \let\@classz\@tabclassz
168 \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}

\@tabarray RmS 91/11/04 added \m@th.
169 \def\@tabarray{\m@th\ifnextchar[\@array{\@array[c]}}

RmS 1993/11/03 changed \halign to \ialign and removed superfluous
\tabskip assignment

\@array
170 \def\@array[#1]#2{%
171 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
172 \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.

173 \setbox\@arstrutbox\hbox{%
174 \vrule \@height\arraystretch\ht\strutbox
175 \depth\arraystretch \dp\strutbox
176 \width\z@}%
177 \mkpream{#2}%
178 \edef\@preamble{%
179 \ialign \noexpand\@halignto
180 \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.

181 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
182 \let\tabularnewline\\%
183 \let\par\@empty
184 \let\@sharp##%
185 \set@typeset@protect
186 \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).

187 \ifhmode \@preamerr\z@ \@@par\fi
188 \@preamble}

\@arraycr Array version of \\.
189 \def\@arraycr{%
190 ${\ifnum0=}\fi\@ifstar\@xarraycr\@arraycr}

```

```

\@arraycr
191 \def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0='{ \fi}$\}\cr}}

\@argarraycr
192 \def\@argarraycr[#1]{%
193   \ifnum0='{ \fi}$\}\ifdim #1>\z@ \@xargarraycr{#1}\else
194     \@yargarraycr{#1}\fi}

\tabularnewline Tabular version of \\.
195 \let\tabularnewline\relax

\@tabularcr
196 \def\@tabularcr{%
197   {\ifnum0='{ \fi}\@ifstar\@xtabularcr\@xtabularcr}

\@xtabularcr
198 \def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0='{ \fi}\cr}}

\@argtabularcr
199 \def\@argtabularcr[#1]{%
200   \ifnum0='{ \fi}%
201   \ifdim #1>\z@
202     \unskip\@xargarraycr{#1}%
203   \else
204     \@yargarraycr{#1}%
205   \fi}

\@xargarraycr
206 \def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
207   \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}

\@yargarraycr
208 \def\@yargarraycr#1{\cr\noalign{\vskip #1}}

\multicolumn Historical LATEX 2.09 comments (not necessarily accurate any more):
\multicolumn{NUMBER}{FORMAT}{ITEM} ==
BEGIN
\multispan{NUMBER}
\begingroup
\@addamp == null
\@mkpream{FORMAT}
\@sharp == ITEM
\protect == \relax
\@startpbox == \@@startpbox
\@endpbox == \@@endpbox
\@arstrut
\@preamble
\endgroup
END

```

End of historical L^AT_EX 2.09 comments.

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.

```
209 \long\def\multicolumn#1#2#3{\multispan{#1}\begingroup
210   \mkpream{#2}%
211   \def\@sharp{#3}\set@typeset@protect
212   \let\@startpbox\@startpbox\let\@endpbox\@endpbox
213   \@arstrut \@preamble\hbox{}\endgroup\ignorespaces}
```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

Codes for classes and character numbers of array, tabular and multicolumn arguments.

Character	Class	Number
c	0	0
l	0	1
r	0	2
l	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 ->                               % l
  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 \@arrayclassiv 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 \@exexpast S *0x\@@ END

\@exexpast 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 == \@exexpast
    else \reserved@b == \@exnoop
    fi
  \expandafter \reserved@b \reserved@a S3 \@@
END
End of historical LATEX 2.09 comments.

\@exnoop
214 \def\@exnoop #1\@@{}

\@expast
215 \def\@expast#1{\@exexpast #1*0x\@@}

\@expast
216 \def\@expast#1*#2#3#4\@@{%
217   \edef\reserved@a{#1}%
218   \@tempcnta#2\relax
219   \ifnum\@tempcnta>\z@
220     \@whilenum\@tempcnta>\z@\do
221       {\edef\reserved@a{\reserved@a#3}\advance\@tempcnta \m@ne}%
222     \let\reserved@b\@expast
223   \else
224     \let\reserved@b\@exnoop
225   \fi
226   \expandafter\reserved@b\reserved@a #4\@@}

\if@firstamp
  \@addamp 227 \newif\if@firstamp
228 \def\@addamp{%
229   \if@firstamp
230     \@firstampfalse
231   \else
232     \edef\@preamble{\@preamble &}%
233   \fi}

```

```

\@arrayacol
\@tabacol 234 \def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\@ampacol 235 \def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\@acolampacol 236 \def\@ampacol{\@addamp \@acol}
237 \def\@acolampacol{\@acol\@addamp\@acol}

\@mkpream
238 \def\@mkpream#1{\@firstampttrue\@lastchclass6
239 \let\@preamble\@empty
240 \let\protect\@unexpandable\protect
241 \let\@sharp\relax
242 \let\@startpbox\relax\let\@endpbox\relax
243 \@expast{#1}%
244 \expandafter\@tfor \expandafter
245 \nextchar \expandafter:\expandafter=\reserved@a\do
246 {\@testpach\@nextchar
247 \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
248 \or \@classiv \or\@classv \fi\@lastchclass\@chclass}%
249 \ifcase \@lastchclass \@acol
250 \or \or \@preamerr \@one\or \@preamerr \tw@\or \or \@acol \fi}

\@arrayclassz
251 \def\@arrayclassz{\ifcase \@lastchclass \@acolampacol \or \@ampacol \or
252 \or \or \@addamp \or
253 \@acolampacol \or \@firstampfalse \@acol \fi
254 \edef\@preamble{\@preamble
255 \ifcase \@chnum
256 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
257 \or \hfil$\relax\@sharp$\fi}}

\@tabclassz RmS 91/08/14 inserted extra braces around entry for NFSS
258 \def\@tabclassz{%
259 \ifcase\@lastchclass
260 \@acolampacol
261 \or
262 \@ampacol
263 \or
264 \or
265 \or
266 \@addamp
267 \or
268 \@acolampacol
269 \or
270 \@firstampfalse\@acol
271 \fi
272 \edef\@preamble{%
273 \@preamble{%
274 \ifcase\@chnum
275 \hfil
276 \hskip1sp%
277 \ignorespaces\@sharp\unskip\hfil
278 \or

```

```

279      \hskip1sp\ignorespaces\@sharp\unskip\hfil
280      \or
281      \hfil\hskip1sp\ignorespaces\@sharp\unskip
282      \fi}}

\@classi
283 \def\@classi{%
284   \ifcase\@lastchclass
285     \@acol\@arrayrule
286   \or
287     \@addtopreamble{\hskip \doublerulesep}\@arrayrule
288   \or
289   \or
290   \or
291     \@arrayrule
292   \or
293     \@acol\@arrayrule
294   \or
295     \@arrayrule
296   \fi}

\@classii
297 \def\@classii{%
298   \ifcase\@lastchclass
299   \or
300     \@addtopreamble{\hskip .5\arrayrulewidth}%
301   \fi}

\@classiii
302 \def\@classiii{\ifcase \@lastchclass \@acolampacol \or
303   \@addamp\@acol \or
304   \or \or \@addamp \or
305   \@acolampacol \or \@ampacol \fi}

\@tabclassiv
306 \def\@tabclassiv{\@addtopreamble\@nextchar}

\@arrayclassiv
307 \def\@arrayclassiv{\@addtopreamble{$\@nextchar$}}

\@classv
308 \def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
309 \@sharp\@endpbox}}

\@addtopreamble
310 \def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}

\@chclass
\@lastchclass 311 \newcount\@chclass
\@chnum 312 \newcount\@lastchclass
313 \newcount\@chnum

```

```

\arraycolsep
\arrayrulewidth
\doublerulesep

\arraystretch

\@arstrutbox
\@arstrut

\arrayrule

\@testpatch

\hline

\@xhline

\vline

\cline
\@cline

```

314 \newdimen\arraycolsep
315 \newdimen\arrayrulewidth
316 \newdimen\doublerulesep
317 \newdimen\doublerulesep

318 \def\arraystretch{1} % Default value.

319 \newbox\@arstrutbox
320 \def\@arstrut{%
321 \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}

322 \def\arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
323 \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}

324 \def\@testpatch#1{\@chclass \ifnum \@lastchclass=\tw@ 4 \else
325 \ifnum \@lastchclass=3 5 \else
326 \z@ \if #1c\@chnum \z@ \else
327 \if #1l\@chnum \@ne \else
328 \if #1r\@chnum \tw@ \else
329 \chclass \if #1\@ne \else
330 \if #1@ \tw@ \else
331 \if #1p3 \else \z@ \@preamerr 0\fi
332 \fi \fi \fi \fi \fi \fi
333 \fi}

334 \def\hline{%
335 \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
336 \reserved@a\@xhline}

337 \def\@xhline{\ifx\reserved@a\hline
338 \vskip\doublerulesep
Measure from the middle of the rules.
339 \vskip-\arrayrulewidth
340 \fi
341 \ifnum0='{ \fi}}

342 \def\vline{\vrule \@width \arrayrulewidth}

The old L^AT_EX 2.09 implementation of \cline used up quite a lot of memory and
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 ltblatex.dtx.
These counters are no longer declared.

\newcount\@cla
\newcount\@clb

```

343 \def\cline#1{\@cline#1\@nil}

344 \def\@cline#1-#2\@nil{%
345   \omit

Use the counter from \multispan.
346   \@multicnt#1%
347   \advance\@multispan\m@ne
348   \ifnum\@multicnt=\@ne\@firstofone{&\omit}\fi
349   \@multicnt#2%
350   \advance\@multicnt-#1%
351   \advance\@multispan\@ne

The original had \unskip at this point, but how could a skip get here ???
352   \leaders\hrule\@height\arrayrulewidth\hfill
353   \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.
354   \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 \multiput.
\sp@n
355 \def\multispan{\omit\@multispan}
356 \def\@multispan#1{%
357   \@multicnt#1\relax
358   \loop\ifnum\@multicnt>\@ne \sp@n\repeat}
359 \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)
360 \def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
361 \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
362 \let\@@startpbox=\@startpbox
363 \let\@@endpbox=\@endpbox

364 </2kernel>

```


File F

ltpictur.dtx

69 Picture Mode

Picture mode commands. In addition to the commands available in L^AT_EX 2.09, This section adds the new `\qbezier` command for drawing curves.

`\qbezier` `\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.

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

<code>\unitlength</code>	= value of dimension argument
<code>\@wholewidth</code>	= current line width
<code>\@halfwidth</code>	= half of current line width
<code>\@linefont</code>	= font for drawing lines
<code>\@circlefont</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.

End of historical L^AT_EX 2.09 comments.

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\def\picture#1{\pictur@#1}
9 \def\pictur@(#1){%
10  \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}

```

```

\@picture
11 \def\@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 \def\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\def\put(#1,#2)#3{%
22   \@killglue\raise#2\unitlength
23   \hb@xt@\z@\{\kern#1\unitlength #3\hss}%
24   \ignorespaces}

\multiput #3 had better be a (.
25 \def\multiput(#1,#2)#3{%
26   \@xdim #1\unitlength
27   \@ydim #2\unitlength
28   \@multiput{ }

\multiput
29 \long\def\@multiput(#1,#2)#3#4{%
30   \@killglue\@multicnt #3\relax
31   \@whilenum \@multicnt >\z@\do
32     {\raise\@ydim\hb@xt@\z@\{\kern\@xdim #4\hss}%
33     \advance\@multicnt\m@ne
34     \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
35   \ignorespaces}

\@killglue
36 \def\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}

\thinlines
\thicklines
37 \DeclareRobustCommand\thinlines{\let\@linefnt\tenln \let\@circlefnt\tencirc
38   \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
39 \DeclareRobustCommand\thicklines{\let\@linefnt\tenlnw \let\@circlefnt\tencircw
40   \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}

\linethickness
41 \DeclareRobustCommand\linethickness[1]
42   {\@wholewidth #1\relax \@halfwidth .5\@wholewidth}

\ishortstack
43 \def\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}

```

```

\@ishortstack
44 \def\@shortstack[#1]{%
45   \leavevmode
46   \vbox\bgroup
47     \baselineskip-\p@\lineskip 3\p@
48     \let\mb@l\hss\let\mb@r\hss
49     \expandafter\let\csname mb@#1\endcsname\relax
50     \let\\\@stackcr
51     \@ishortstack}

\@ishortstack
52 \def\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\cr}\egroup}

\@stackcr
\@ixstackcr
53 \def\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
54 \def\@ixstackcr{\@ifnextchar[\@istackcr{\cr\ignorespaces}}

\@istackcr
55 \def\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}

Historical LATEX 2.09 comments (not necessarily accurate any more):
\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
  fi
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

```

End of historical L^AT_EX 2.09 comments.

\if@negarg

56 \newif\if@negarg

\line

```

57 \def\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
58   \@linelen #3\unitlength
59   \ifdim\@linelen<\z@\@badlinearg\else
60     \ifnum\@xarg =\z@ \@vline
61     \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
62   \fi
63 \fi}

```

\@sline

```

64 \def\@sline{%
65   \ifnum\@xarg<\z@ \@negargtrue \@xarg -\@xarg \@yyarg -\@yarg
66   \else \@negargfalse \@yyarg \@yarg \fi
67   \ifnum \@yyarg >\z@ \@tempcnta\@yyarg \else \@tempcnta -\@yyarg \fi
68   \ifnum\@tempcnta>6 \@badlinearg\@tempcnta\z@ \fi
69   \ifnum\@xarg>6 \@badlinearg\@xarg \@ne \fi
70   \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.

```

71 \ifdim\wd\@linechar=\z@
72   \setbox\@linechar\hbox{.}%
73   \@badlinearg
74 \fi
75 \ifnum \@yarg >\z@ \let\@upordown\raise \@clnht\z@
76   \else\let\@upordown\lower \@clnht \ht\@linechar\fi
77 \@clnwd \wd\@linechar
78 \if@negarg
79   \hskip -\wd\@linechar \def\reserved@a{\hskip -2\wd\@linechar}%
80 \else
81   \let\reserved@a\relax
82 \fi
83 \@whiledim \@clnwd <\@linelen \do
84   {\@upordown\@clnht\copy\@linechar
85     \reserved@a
86     \advance\@clnht \ht\@linechar
87     \advance\@clnwd \wd\@linechar}%
88 \advance\@clnht -\ht\@linechar
89 \advance\@clnwd -\wd\@linechar
90 \@tempdima\@linelen\advance\@tempdima -\@clnwd
91 \@tempdimb\@tempdima\advance\@tempdimb -\wd\@linechar
92 \if@negarg \hskip -\@tempdimb \else \hskip \@tempdimb \fi
93 \multiply\@tempdima \@m
94 \@tempcnta \@tempdima
95 \@tempdima \wd\@linechar \divide\@tempcnta \@tempdima
96 \@tempdima \ht\@linechar \multiply\@tempdima \@tempcnta
97 \divide\@tempdima \@m
98 \advance\@clnht \@tempdima
99 \ifdim \@linelen <\wd\@linechar
100   \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).

```

101 \ifdim \@linelen = \z@
102 \else
103   \@picture@warn
104 \fi
105 \else\@upordown\@clnht\copy\@linechar\fi}

```

\@hline

```

106 \def\@hline{\ifnum \@xarg <\z@ \hskip -\@linelen \fi
107 \vrule \@height \@halfwidth \@depth \@halfwidth \@width \@linelen
108 \ifnum \@xarg <\z@ \hskip -\@linelen \fi}

```



```

\@getlinechar
109 \def\@getlinechar(#1,#2){\@tempcnta#1\relax\multiply\@tempcnta 8%
110 \advance\@tempcnta -9\ifnum #2>\z@ \advance\@tempcnta #2\relax\else
111 \advance\@tempcnta -#2\relax\advance\@tempcnta 64 \fi
112 \char\@tempcnta}

\vector
113 \def\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
114 \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
115 \ifnum\@tempcnta<5\relax
116 \@linelen #3\unitlength
117 \ifdim\@linelen<\z@\@badlinearg\else
118 \ifnum\@xarg =\z@ \vvector
119 \else \ifnum\@yarg =\z@ \hvector \else \svector\fi
120 \fi
121 \fi
122 \else\@badlinearg\fi}

\hvector
123 \def\hvector{\@hline\hb@xt@\z@{\@linefnt
124 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
125 \hss\@getrarrow(1,0)\fi}}

\vvector
126 \def\vvector{\ifnum \@yarg <\z@ \downvector \else \upvector \fi}

\svector
127 \def\svector{\@sline
128 \@tempcnta\@yarg \ifnum\@tempcnta <\z@ \@tempcnta -\@tempcnta\fi
129 \ifnum\@tempcnta <5%
130 \hskip -\wd\@linechar
131 \upordown\@clnht \hbox{\@linefnt \if@negarg
132 \@getlarrow(\@xarg,\@yyarg)\else \@getrarrow(\@xarg,\@yyarg)\fi}%
133 \else\@badlinearg\fi}

\@getlarrow
134 \def\@getlarrow(#1,#2){\ifnum #2=\z@ \@tempcnta 27 % '33
135 \else
136 \@tempcnta #1\relax\multiply\@tempcnta \sixt@@n
137 \advance\@tempcnta -9 \@tempcntb #2\relax\multiply\@tempcntb \tw@
138 \ifnum \@tempcntb >\z@ \advance\@tempcnta \@tempcntb
139 \else\advance\@tempcnta -\@tempcntb\advance\@tempcnta 64
140 \fi\fi\char\@tempcnta}

\@getrarrow
141 \def\@getrarrow(#1,#2){\@tempcntb #2\relax
142 \ifnum\@tempcntb <\z@ \@tempcntb -\@tempcntb\relax\fi
143 \ifcase \@tempcntb\relax \@tempcnta 45 % '55
144 \or
145 \ifnum #1<\thr@@ \@tempcnta #1\relax\multiply\@tempcnta
146 24\advance\@tempcnta -6 \else \ifnum #1=\thr@@ \@tempcnta 49
147 \else\@tempcnta 58 \fi\fi\or

```

```

148 \ifnum #1<\thr@@ \@tempcnta=#1\relax\multiply\@tempcnta
149 24\advance\@tempcnta -\thr@@ \else \@tempcnta 51 \fi\or
150 \@tempcnta #1\relax\multiply\@tempcnta
151 \sift@@n \advance\@tempcnta -\tw@ \else
152 \@tempcnta #1\relax\multiply\@tempcnta
153 \sift@@n \advance\@tempcnta 7 \fi\ifnum #2<\z@ \advance\@tempcnta 64 \fi
154 \char\@tempcnta}

\@vline
155 \def\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}

\@upline
156 \def\@upline{%
157   \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
158     \@height \@linelen \@depth \z@\hss}}

\@downline
159 \def\@downline{%
160   \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
161     \@height \z@ \@depth \@linelen \hss}}

\@upvector
162 \def\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}% '66
163   \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}

\@downvector
164 \def\@downvector{\@downline\lower \@linelen
165   \hb@xt@\z@{\@linefnt\char 63 % '77
166   \hss}}

Historical LATEX 2.09 comments (not necessarily accurate any more):
\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
End of historical LATEX 2.09 comments.

\dashbox
167 \def\dashbox#1(#2,#3){\leavevmode\hb@xt@\z@{\baselineskip \z@skip
168 \lineskip \z@skip
169 \@dashdim #2\unitlength
170 \@dashcnt \@dashdim \advance\@dashcnt 200
171 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
172 \ifodd\@dashcnt\@dashdim \z@
173 \advance\@dashcnt \@one \divide\@dashcnt \tw@
174 \else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
175 \advance\@dashcnt \m@ne
176 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
177 \@width \@dashdim}\put(0,0){\copy\@dashbox}%
178 \put(0,#3){\copy\@dashbox}%
179 \put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
180 \put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
181 \multiply\@dashdim \thr@@
182 \fi
183 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
184 \@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
185 \put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
186 \do{\copy\@dashbox\advance\@tempcnta \@one }}\@tempcnta\z@
187 \put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
188 \do{\copy\@dashbox\advance\@tempcnta \@one }}%
189 \@dashdim #3\unitlength
190 \@dashcnt \@dashdim \advance\@dashcnt 200
191 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
192 \ifodd\@dashcnt \@dashdim \z@
193 \advance\@dashcnt \@one \divide\@dashcnt \tw@
194 \else
195 \divide\@dashdim \tw@ \divide\@dashcnt \tw@
196 \advance\@dashcnt \m@ne
197 \setbox\@dashbox\hbox{\hskip -\@halfwidth
198 \vrule \@width \@wholewidth
199 \@height \@dashdim}\put(0,0){\copy\@dashbox}%

```

```

200 \put(#2,0){\copy\@dashbox}%
201 \put(0,#3){\lower\@dashdim\copy\@dashbox}%
202 \put(#2,#3){\lower\@dashdim\copy\@dashbox}%
203 \multiply\@dashdim \thr@@
204 \fi
205 \setbox\@dashbox\hbox{\vrule \@width \@wholewidth
206 \@height #1\unitlength}\@tempcnta\z@
207 \put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
208 \do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
209 \vskip\@dashdim}}\@tempcnta\z@
210 \put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
211 \do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
212 \vskip\@dashdim}}}\@makepicbox(#2,#3)}

```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

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 `\@ovvxx`.

`\@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
End of historical LATEX 2.09 comments.

```



```

\if@ovt If producing the Top Bottom Left or Right of an oval.
\if@ovb 213 \newif\if@ovt
\if@ovl 214 \newif\if@ovb
\if@ovr 215 \newif\if@ovl
        216 \newif\if@ovr

\@ovxx
\@ovyy 217 \newdimen\@ovxx
\@ovdx 218 \newdimen\@ovyy
\@ovdy 219 \newdimen\@ovdx
\@ovro 220 \newdimen\@ovdy
\@ovri 221 \newdimen\@ovro
        222 \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
223 \def\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
224 \@tempcnta\@tempdima
225 \@tempdima 4\p@ \divide\@tempcnta\@tempdima
226 \ifnum \@tempcnta >10\relax
227 \@picture@warn
228 \@tempcnta 10\relax
229 \fi
230 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
Warn if requirements for oval or circle can't be met.
231 \else \@picture@warn \fi
232 \multiply\@tempcnta 4\relax
233 \setbox \@tempboxa \hbox{\@circlefnt
234 \char \@tempcnta}\@tempdima \wd \@tempboxa}

\@picture@warn Generic warning for lines, vectors (used in \@sline) and oval or circle (used in
\@getcirc) are not available at right size.
235 \def\@picture@warn{\@latex@warning{%
236 \string\oval, \string\circle, or \string\line\space
237 size unavailable}}

\@put
238 \def\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}

\oval
239 \def\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2) []}}
240 \if2\kernel
241 \if\latexrelease\IncludeInRelease{2016/03/31}%
242 \if\latexrelease \if\@ovhlinetrue%
243 \if\latexrelease {Avoid almost zero length leaders}%
244 \if2\kernel\if\latexrelease

\if@ovvline Tests whether horizontal or vertical lines are needed.
\if@ovhline 245 \newif\if@ovvline \if@ovvlinetrue
246 \newif\if@ovhline \if@ovhlinetrue

```

\@oval

```

247 \def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
248   \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue

249   \@ovvlinefalse \@ovhlinefalse

250   \@tfor\reserved@a :=#3\do{\csname @ov\reserved@a false\endcsname}%
251   \@ovxx #1\unitlength
252   \@ovyy #2\unitlength

253   \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
254   \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue \fi\fi

255   \advance \@tempdimb -2\p@
256   \@getcirc \@tempdimb
257   \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
258   \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
259   \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@

260   \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
261   \ifdim \@ovdy >\z@ \@ovvlinetrue \fi

262   \@circlefnt \setbox\@tempboxa
263   \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
264   \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
265   \if@ovt \@ovhorz \kern -\@ovxx \fi
266   \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
267   \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
268   \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
269   \endgroup}

```

\@ovvert

```

270 \def\@ovvert#1#2{\vbox to\@ovyy{%
271   \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
272   \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
273   \else \kern \@ovri \kern \@ovdy \fi

274   \if@ovvline \leaders\vrule \@width \@wholewidth \fi

275   \vfil \nointerlineskip
276   \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
277   \hbox{\char \@tempcntb}%
278   \else \kern \@ovdy \kern \@ovro \fi}}

```

\@ovhorz

```

279 \def\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
280   \if@ovr \else \kern \@ovdx \fi

281   \if@ovhline \leaders \hrule \@height \@wholewidth \fi

282   \hfil
283   \if@ovl \else \kern \@ovdx \fi
284   \kern \@ovri}}

285 \if2ekernel\latexrelease\
286 \if2ekernel\EndIncludeInRelease
287 \if2ekernel\IncludeInRelease{0000/00/00}%
288 \if2ekernel\{\@ovhlinetrue}%

```

```

289 \latexrelease\{Avoid almost zero length leaders}%
290 \latexrelease\let\if@ovvline\undefined
291 \latexrelease\let\if@ovhline\undefined
292 \latexrelease\def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
293 \latexrelease\@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
294 \latexrelease\@tfor\reserved@a :=#3\do
295 \latexrelease\{\csname @ov\reserved@a false\endcsname}%
296 \latexrelease\@ovxx #1\unitlength
297 \latexrelease\@ovyy #2\unitlength
298 \latexrelease\@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
299 \latexrelease\advance \@tempdimb -2\p@
300 \latexrelease\@getcirc \@tempdimb
301 \latexrelease\@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
302 \latexrelease\@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
303 \latexrelease\@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
304 \latexrelease\@circlefnt \setbox\@tempboxa
305 \latexrelease\hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
306 \latexrelease\if@ovl
307 \latexrelease\kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
308 \latexrelease\fi
309 \latexrelease\if@ovt \@ovhorz \kern -\@ovxx \fi
310 \latexrelease\if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
311 \latexrelease\advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
312 \latexrelease\@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
313 \latexrelease\endgroup}
314 \latexrelease\def\@ovvert#1#2{\vbox to\@ovyy{%
315 \latexrelease\if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
316 \latexrelease\kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
317 \latexrelease\else \kern \@ovri \kern \@ovdy \fi
318 \latexrelease\leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
319 \latexrelease\if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
320 \latexrelease\hbox{\char \@tempcntb}%
321 \latexrelease\else \kern \@ovdy \kern \@ovro \fi}}
322 \latexrelease\def\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
323 \latexrelease\if@ovr \else \kern \@ovdx \fi
324 \latexrelease\leaders \hrule \@height \@wholewidth \hfil
325 \latexrelease\if@ovl \else \kern \@ovdx \fi
326 \latexrelease\kern \@ovri}}
327 \latexrelease\EndIncludeInRelease
328 \*2kernel)

```

\circle

```

329 \def\circle{\@inmatherr\circle\@ifstar\@dot\@circle}

```

\@circle

```

330 \def\@circle#1{%
331 \begingroup \boxmaxdepth \maxdimen \@tempdimb #1\unitlength
332 \ifdim \@tempdimb >15.5\p@ \@getcirc\@tempdimb
333 \ovro\ht\@tempboxa
334 \setbox\@tempboxa\hbox{\@circlefnt
335 \advance\@tempcnta\tw@ \char \@tempcnta
336 \advance\@tempcnta@m@ne \char \@tempcnta \kern -2\@tempdima
337 \advance\@tempcnta\tw@
338 \raise \@tempdima \hbox{\char\@tempcnta}\raise \@tempdima

```

```

339      \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
340      \put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
341      \else \@circ\@tempdimb{96}\fi\endgroup}

\@dot Internal form of \circle*.
342 \def\@dot#1{\@tempdimb #1\unitlength \@circ\@tempdimb{112}}

\@circ
343 \def\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
344 \@tempcnta\@tempdima \@tempdima \p@
345 \divide\@tempcnta\@tempdima
346 \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
347 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
348 \advance\@tempcnta #2\relax
349 \@circlefnt \char\@tempcnta}

\@xarg Counters used for manipulating the ‘slope’ arguments.
\@yarg 350 \newcount\@xarg
\@yyarg 351 \newcount\@yarg
352 \newcount\@yyarg

\@multicnt Counter used in \multiput, and also \multicolumn.
353 \newcount\@multicnt

\@xdim Length registers.
\@ydim 354 \newdimen\@xdim
355 \newdimen\@ydim

\@linechar Box for holding a line segment character, for sloping lines.
356 \newbox\@linechar

\@linelen Length of the line currently being built.
357 \newdimen\@linelen

\@clnwd Height and width of current line segment.
\@clnht 358 \newdimen\@clnwd
359 \newdimen\@clnht

\@dashdim \dashbox internal registers.
\@dashbox 360 \newdimen\@dashdim
\@dashcnt 361 \newbox\@dashbox
362 \newcount\@dashcnt

Initialization: “\thinline”
363 \let\@linefnt\tenln
364 \let\@circlefnt\tencirc
365 \@wholewidth\fontdimen8\tenln
366 \@halfwidth .5\@wholewidth

```

69.1 Curves

The new `\qbezier` command, based on the old `\bezier` defined in `bezier.sty`.
Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```
\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
```

End of historical L^AT_EX 2.09 comments.

`\qbeziermax` The maximum number of points to plot.
 367 `\def\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 (.
368 \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 (.
369 \def\bezier#1)#2(#3)#4({\@bezier#1)(#3){}

\@bezier
370 \def\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
371   \ifnum #1=\z@
372     \@ovxx #4\unitlength
373     \advance\@ovxx -#2\unitlength
374     \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
375     \@ovdx #6\unitlength
376     \advance\@ovdx -#4\unitlength
377     \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
378     \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
379     \@ovyy #5\unitlength
380     \advance\@ovyy -#3\unitlength
381     \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
382     \@ovdy #7\unitlength
383     \advance\@ovdy -#5\unitlength
384     \ifdim \@ovdy<\z@ \@ovdy -\@ovdy \fi
385     \ifdim \@ovyy<\@ovdy \@ovyy \@ovdy \fi
386     \@multicnt
387     \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
388     \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
389     \ifnum \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax \fi
390   \else \@multicnt#1\relax \fi
391   \@tempcnta\@multicnt \advance\@tempcnta\@ne
392   \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
393   \multiply\@ovdx \tw@
394   \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
395   \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
396   \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
397   \multiply\@ovdy \tw@
398   \@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
399   \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt

400   \setbox\@tempboxa\hbox{%
401     \hskip -\@halfwidth
402     \vrule \@height\@halfwidth
403     \@depth \@halfwidth
404     \@width \@wholewidth}%
405   \put(#2,#3){%
406     \count@\z@
407     \@whilenum{\count@<\@tempcnta}\do
408       {\@xdim\count@\@ovxx
409        \advance\@xdim\@ovdx

```

```

410          \divide\@xdim\@multicnt
411          \multiply\@xdim\count@
412          \@ydim\count@\@ovvy
413          \advance\@ydim\@ovdy
414          \divide\@ydim\@multicnt
415          \multiply\@ydim\count@
416          \raise \@ydim
417          \hb@xt@\z@{\kern\@xdim
418              \unhcopy\@tempboxa\hss}%
419          \advance\count@\@ne}}

```

As the commands above all use “picture” interface we couldn’t define them with `\DeclareRobustCommand` so we do that now.

```

420 \</2kernel>
421 \<*2kernel | latexrelease>
422 \<latexrelease>\IncludeInRelease{2019/10/01}%
423 \<latexrelease>          {\bezier}{Make commands robust}%
424 \MakeRobust\bezier
425 \MakeRobust\circle
426 \MakeRobust\dashbox
427 \MakeRobust\line
428 \MakeRobust\linethickness
429 \MakeRobust\multiput
430 \MakeRobust\oval
431 \MakeRobust\put
432 \MakeRobust\qbezier
433 \MakeRobust\shortstack
434 \MakeRobust\thinline
435 \MakeRobust\vector
436 \</2kernel | latexrelease>
437 \<latexrelease>\EndIncludeInRelease
438 \<latexrelease>\IncludeInRelease{0000/00/00}%
439 \<latexrelease>          {\bezier}{Make commands robust}%
440 \<latexrelease>
441 \<latexrelease>\kernel@make@fragile\bezier
442 \<latexrelease>\kernel@make@fragile\circle
443 \<latexrelease>\kernel@make@fragile\dashbox
444 \<latexrelease>\kernel@make@fragile\line
445 \<latexrelease>\kernel@make@fragile\linethickness
446 \<latexrelease>\kernel@make@fragile\multiput
447 \<latexrelease>\kernel@make@fragile\oval
448 \<latexrelease>\kernel@make@fragile\put
449 \<latexrelease>\kernel@make@fragile\qbezier
450 \<latexrelease>\kernel@make@fragile\shortstack
451 \<latexrelease>\kernel@make@fragile\thinline
452 \<latexrelease>\kernel@make@fragile\vector
453 \<latexrelease>
454 \<latexrelease>\EndIncludeInRelease
455 \<*2kernel>
456 \</2kernel>

```

File G

ltthm.dtx

70 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 $\text{\the}\langle name \rangle$.

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

DOCUMENT STYLE PARAMETERS

$\text{\@thmcounter}\{COUNTER\}$: A command such that

```
\edef\theCOUNTER{\@thmcounter{COUNTER}}
```

defines $\text{\the}COUNTER$ to produce a number for a theorem environment.

The default is:

```
BEGIN \noexpand\arabic{COUNTER} END
```

\@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, \@thmcountersep should be \def ’ed to ‘-’. Its default is ‘.’.

$\text{\@begintheorem}\{NAME\}\{NUMBER\}$: A command that begins a theorem

environment for a ‘theorem’ named ‘NAME NUMBER’ – e.g., $\text{\@begintheorem}\{Lemma\}\{3.7\}$ starts Lemma 3.7.

$\text{\@opargbegintheorem}\{NAME\}\{NUMBER\}\{OPARG\}$:

A command that begins a theorem environment for a ‘theorem’ named ‘NAME NUMBER’ with

optional

argument OPARG – e.g., $\text{\@begintheorem}\{Lemma\}\{3.7\}\{Jones\}$ starts ‘Lemma 3.7 (Jones)’.

\@endtheorem : A command that ends a theorem environment.


```

\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
End of historical LATEX 2.09 comments.

\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 LATEX 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 \endkernel
```

File H

ltsect.dtx

71 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^AT_EX 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,}
```

71.1 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.

3 \*2ekernel)
4 \*2ekernel | latexrelease)
5 (latexrelease)\IncludeInRelease{2019/10/01}%
6 (latexrelease) {\title}{Make commands robust}%

\title \title for use in \maketitle. If not given \maketitle will produce an error
message.
7 \DeclareRobustCommand\title[1]{\gdef\@title{#1}}

\author \author for use in \maketitle. If not given \maketitle will produce a warning
message.
8 \DeclareRobustCommand\author[1]{\gdef\@author{#1}}

\date \date for use in \maketitle. If not given \maketitle will produce \today as the
default.
9 \DeclareRobustCommand\date[1]{\gdef\@date{#1}}

\thanks
10 \DeclareRobustCommand\thanks[1]{\footnotemark
11 \protected@xdef\@thanks{\@thanks
12 \protect\footnotetext[\the\c@footnote]{#1}}%
13 }
```

```

\and
14 \DeclareRobustCommand\and{%    % \begin{tabular}
15   \end{tabular}}%
16   \hskip 1em \@plus.17fil%
17   \begin{tabular}[t]{c}}%      % \end{tabular}

18 </2ekernel | latexrelease>
19 <latexrelease>\EndIncludeInRelease
20 <latexrelease>\IncludeInRelease{0000/00/00}%
21 <latexrelease>                {\title}{Make commands robust}%
22 <latexrelease>
23 <latexrelease>\kernel@make@fragile\title
24 <latexrelease>\kernel@make@fragile\author
25 <latexrelease>\kernel@make@fragile\date
26 <latexrelease>\kernel@make@fragile\thanks
27 <latexrelease>\kernel@make@fragile\and
28 <latexrelease>
29 <latexrelease>\EndIncludeInRelease
30 <*2ekernel>

\@title
31 \def\@title{\@latex@error{No \noexpand\title given}\@ehc}

\@author
32 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}

\@date
33 \gdef\@date{\today}

\@thanks
34 \let\@thanks\@empty
35 \message{sectioning,}

```

71.2 Sectioning

```

\@secpenalty
36 \newcount\@secpenalty
37 \@secpenalty = -300

\if@noskipsec \@noskipsectrue
Way back in 1991 (08/26) FMi & RmS set the \@noskipsec switch to true for the
preamble and to false in \document. This was done to trap lists and related text
in the preamble but it does not catch everything.
38 \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 = 0, section = 1, etc.

indent: Indentation of heading from left margin

beforeskip: Absolute value = skip to leave above the heading. If negative, then paragraph indent of text following heading is suppressed.

afterskip: 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 *Historical L^AT_EX 2.09 comments (not necessarily accurate any more):*

```
\@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
```

End of historical L^AT_EX 2.09 comments.

```

39 \def\@startsection#1#2#3#4#5#6{%
40   \if@noskipsec \leavevmode \fi
41   \par
42   \@tempskipa #4\relax
43   \@afterindenttrue
44   \ifdim \@tempskipa <\z@
45     \@tempskipa -\@tempskipa \@afterindentfalse
46   \fi
47   \if@nobreak
48     \everypar{}%
49   \else
50     \addpenalty\@secpenalty\addvspace\@tempskipa
51   \fi
52   \ifstar
53     {\@ssect{#3}{#4}{#5}{#6}}%
54     {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

`\@sect` Pseudocode for the `\@sect` command *Historical L^AT_EX 2.09 comments (not necessarily accurate any more):*

```

\@sect{NAME}{LEVEL}
  {INDENT}{BEFORESKIP}{AFTERSKIP}
  {STYLE}[ARG1]{ARG2}
  ==
BEGIN
  IF LEVEL > \c@secnumdepth
    THEN \@svsec :=L null
    ELSE \refstepcounter{NAME}
         \@svsec :=L BEGIN \@secntformat{#1}\relax END
  FI
  IF AFTERSKIP > 0
    THEN \begingroup
         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

End of historical L^AT_EX 2.09 comments.

```
55 \def\@sect#1#2#3#4#5#6[#7]#8{%
56   \ifnum #2>\c@secnumdepth
57     \let\@svsec\@empty
58   \else
59     \refstepcounter{#1}%
```

Since \@secntformat 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.

```
60   \protected@edef\@svsec{\@secntformat{#1}\relax}%
61   \fi
62   \@tempskipa #5\relax
63   \ifdim \@tempskipa>\z@
64     \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 #6.

```
65     #6{%
66       \@hangfrom{\hskip #3\relax\@svsec}%
67       \interlinepenalty \@M #8\@par}%
68   \endgroup
69   \csname #1mark\endcsname{#7}%
70   \addcontentsline{toc}{#1}{%
71     \ifnum #2>\c@secnumdepth \else
72       \protect\numberline{\csname the#1\endcsname}%
73     \fi
74     #7}%
75   \else
```

\relax added 2 May 90

```
76   \def\@svsechd{%
77     #6{\hskip #3\relax
78       \@svsec #8}%
79     \csname #1mark\endcsname{#7}%
80     \addcontentsline{toc}{#1}{%
81       \ifnum #2>\c@secnumdepth \else
82         \protect\numberline{\csname the#1\endcsname}%
83       \fi
84       #7}}%
85   \fi
86   \@xsect{#5}}
```

\@xsect Pseudocode for the \@xsect command *Historical L^AT_EX 2.09 comments (not necessarily accurate any more):*

```
\@xsect{AFTERSKIP} ==
BEGIN
  IF AFTERSKIP > 0
  THEN \par \nobreak
```



```

        \vskip AFTERSKIP
        \@afterheading
ELSE @nobreake :=G F
    @noskipsec :=G T
    \everypar{ IF @noskipsec = T
        THEN @noskipsec :=G F
            \clubpenalty := 10000 % local
            \hskip -\parindent
            \begingroup
            \@svsechd
            \endgroup
            \unskip
            \hskip -AFTERSKIP \relax
            %% relax added 14 Jan 91
        ELSE \clubpenalty := \@clubpenalty % local
            \everypar := NULL
        FI
    }
FI

END

```

End of historical L^AT_EX 2.09 comments.

```

87 \def\xsect#1{%
88   \@tempkipa #1\relax
89   \ifdim \@tempkipa>\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.

```

90   \par \nobreake
91   \vskip \@tempkipa
92   \@afterheading
93   \else

94   \@nobreakefalse
95   \global\noskipsecttrue
96   \everypar{%
97     \if@noskipsec
98       \global\noskipsectfalse
99       {\setbox\z@\lastbox}%
100      \clubpenalty\@M
101      \begingroup \@svsechd \endgroup
102      \unskip
103      \@tempkipa #1\relax
104      \hskip -\@tempkipa
105    \else
106      \clubpenalty \@clubpenalty
107      \everypar{}%
108    \fi}%
109   \fi
110   \ignorespaces}

```

```

\@secntformat This command formats the section number including the space following it.
111 \def\@secntformat#1{\csname the#1\endcsname\quad}

Pseudocode for the \@ssect command Historical LATEX 2.09 comments (not necessarily accurate any more):
\@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
End of historical LATEX 2.09 comments.

Pseudocode for the \@afterheading command Historical LATEX 2.09 comments (not necessarily accurate any more):
\@afterheading ==
BEGIN
  @nobreak :=G true
  \everypar := BEGIN IF @nobreak = T
                    THEN @nobreak :=G false
                      \clubpenalty := 10000 % local
                      IF @afterindent = F
                        THEN remove \lastbox
                      FI
                    ELSE \clubpenalty := \@clubpenalty % local
                      \everypar := NULL
                    FI
  END
END
End of historical LATEX 2.09 comments.

\@ssect
112 \def\@ssect#1#2#3#4#5{%
113   \@tempskipa #3\relax
114   \ifdim \@tempskipa>\z@
115     \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.
116     #4{%
117       \@hangfrom{\hskip #1}%
118       \interlinepenalty \@M #5\@@par}%
119     \endgroup
120   \else

```

```

121     \def\@svsechd{#4{\hskip #1\relax #5}}%
122     \fi
123     \@xsect{#3}}

\if@afterindent
\@afterindenttrue 124 \newif\if@afterindent \@afterindenttrue

\@afterheading This hook is used in setting up custom-built headings in classes.dtx.
125 \def\@afterheading{%
126     \@nobreaktrue
127     \everypar{%
128         \if@nobreak
129             \@nobreakfalse
130             \clubpenalty \@M
131             \if@afterindent \else
132                 {\setbox\z@\lastbox}%
133                 \fi
134             \else
135                 \clubpenalty \@clubpenalty
136                 \everypar{}%
137                 \fi}}

\@hangfrom \@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.
138 \def\@hangfrom#1{\setbox\@tempboxa\hbox{#1}}%
139     \hangindent \wd\@tempboxa\noindent\box\@tempboxa}

\c@secnumdepth
\c@tocdepth 140 \newcount\c@secnumdepth
141 \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*{...}

142 \def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

71.2.1 Initializations

```

\sectionmark
\subsectionmark 143 \let\sectionmark\@gobble
\subsubsectionmark 144 \let\subsectionmark\@gobble
\paragraphmark 145 \let\subsubsectionmark\@gobble
\subparagraphmark 146 \let\paragraphmark\@gobble
147 \let\subparagraphmark\@gobble

148 \message{contents,}
```

71.3 Table of Contents etc.

71.3.1 Convention

`\tf@{foo}` = file number for output for table foo. The file is opened only if `@filesw = true`.

71.3.2 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.

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```
\@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
```

End of historical L^AT_EX 2.09 comments.

```
149 \def\@starttoc#1{%
150   \begingroup
151     \makeatletter
152     \@input{\jobname.#1}%
153     \if@filesw
154       \expandafter\newwrite\csname tf@#1\endcsname
155       \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
156     \fi
157     \@nobreakfalse
158   \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`.

```

159 </2ekernel>
160 <*2ekernel | latexrelease>
161 <latexrelease>\IncludeInRelease{2018/12/01}%
162 <latexrelease>          {\addcontentsline}{Mask line endings}%
163 \def\addcontentsline#1#2#3{%
164   \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}%
We add \protected@file@percent at the end which is turned inside \@writefile
into a percent character to mask the newline after the closing argument brace.
165           \protected@file@percent}}
166 </2ekernel | latexrelease>
167 <latexrelease>\EndIncludeInRelease
168 <latexrelease>\IncludeInRelease{0000/00/00}%
169 <latexrelease>          {\addcontentsline}{Mask line endings}%
170 <latexrelease>\def\addcontentsline#1#2#3{%
171 <latexrelease>   \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}
172 <latexrelease>\EndIncludeInRelease
173 <*2ekernel>

```

`\addtocontents` The `\addtocontents{<table>}{<text>}` command adds `<text>` to the `.<table>` file, with no page number.

```

174 \long\def\addtocontents#1#2{%
175   \protected@write\@auxout
176     {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
177     {\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.

```

178 \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`

```

179 </2kernel>
180 (*2kernel | latexrelease)
181 (latexrelease)\IncludeInRelease{2018/12/01}%
182 (latexrelease)          {\@dottedtocline}{Prevent protrusion}%
183 \def\@dottedtocline#1#2#3#4#5{%
184   \ifnum #1>\c@tocdepth \else
185     \vskip \z@ \@plus.2\p@
186     {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
187      \parindent #2\relax\@afterindenttrue
188      \interlinepenalty\@M
189      \leavevmode
190      \@tempdima #3\relax
191      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
192      {#4}\nobreak
193      \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.

```

194       \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
195       mu$}\hfill
196     \nobreak
197     \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5%
```

We finish off by preventing any protrusion if that is enabled. If protrusion happens the number may shift to the right and as a result you may end up with an additional dot in the toc line in some situations.

```

198                               \kern-\p@\kern\p@}%
199     \par}%
200   \fi}
```

`\noptrusion` This command, if placed directly to the right (or left) of a word, will prevent prevent protrusion of that word into the margin. It is used in the toc entry lines as they shouldn't protrude. It is implemented as to kerns that cancel each other

but being there hide the word so that protrusion is not added. Note that a zero kern or an empty box would not work as the protrusion mechanism will skip over those.

```

201 \DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}

202 \if2kernel | latexrelease)
203 (latexrelease)\EndIncludeInRelease
204 (latexrelease)\IncludeInRelease{0000/00/00}%
205 (latexrelease)          {\@dottedtocline}{Prevent protrusion}%
206 (latexrelease)\def\@dottedtocline#1#2#3#4#5{%
207 (latexrelease)  \ifnum #1>\c@tocdepth \else
208 (latexrelease)    \vskip \z@ \@plus.2\p@
209 (latexrelease)    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
210 (latexrelease)    \parindent #2\relax\@afterindenttrue
211 (latexrelease)    \interlinepenalty\@M
212 (latexrelease)    \leavevmode
213 (latexrelease)    \@tempdima #3\relax
214 (latexrelease)    \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
215 (latexrelease)    {#4}\nobreak
216 (latexrelease)    \leaders\hbox{$\m@th
217 (latexrelease)      \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
218 (latexrelease)      mu$}\hfill
219 (latexrelease)    \nobreak
220 (latexrelease)    \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
221 (latexrelease)    \par}%
222 (latexrelease)  \fi}
223 (latexrelease)
224 (latexrelease)\let\noprotrusion\undefined
225 (latexrelease)\EndIncludeInRelease
226 \if2kernel)

```

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.)

```

227 \def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
228 \if2kernel)

```

File I

lfloat.dtx

72 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.

72.1 Floating Environments

```
1 \*2ekernel
2 \message{floats,}
```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

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`.
End of historical L^AT_EX 2.09 comments.

`\@floatpenalty`

```
3 \newcount\@floatpenalty
```

`\caption` This is set to be an error message outside a float since no `capttype` is defined there; this may need to be changed by some classes.

```
4 \def\caption{%
5   \ifx\@capttype\undefined
6     \latexerror{\noexpand\caption outside float}\@ehd
7     \expandafter\@gobble
8   \else
9     \refstepcounter\@capttype
10    \expandafter\@firstofone
11  \fi
12  {\@dblarg{\@caption\@capttype}}%
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^AT_EX constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates T_EX'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 \if2kernel
35 \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
36 \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
37 \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
38 \def\@xfloat #1[#2]{%
39   \nodocument
40   \def \@capttype {#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 \*2ekernel)

```

`\@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       \ifnum\@floatpenalty <-\@Mii
216         \penalty -\@Miv
217         \@tempdima\prevdepth
218         \vbox{}%
219         \prevdepth\@tempdima
220         \penalty\@floatpenalty
221       \else
222         \vadjust{\penalty -\@Miv \vbox{}}\penalty\@floatpenalty\@Esphack
223       \fi
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 <latexrelease>    \@largefloatcheck
239 <latexrelease>    \@cons\@dbldeferlist\@currbox
240 <latexrelease>    \fi
241 <latexrelease>    \ifnum \@floatpenalty ==-\@Mii \@Esphack\fi
242 <latexrelease>\else
243 <latexrelease>  \end@float
244 <latexrelease>\fi
245 <latexrelease>}%
246 <latexrelease>\EndIncludeInRelease
247 <*2ekernel>
```

We make sure that we never exceed \textheight, otherwise float will never get typeset (91/03/15 FMi).

RmS 92/03/18 changed \@esphack to \@Esphack.


```

\@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 }

\outer@nobreak
255 \let\outer@nobreak\@empty

\@largefloatcheck This calculates by how much a float is oversize for the page and prints this in a
                    warning message.
256 \def \@largefloatcheck{%
257     \ifdim \ht\@currbox>\textheight
258         \@tempdima -\textheight
259         \advance \@tempdima \ht\@currbox
260         \@latex@warning {Float too large for page by \the\@tempdima}%
261         \ht\@currbox \textheight
262     \fi
263 }

\@dbflt
\@xdblfloat 264 \def\@dbflt#1{\@ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
265 \def\@xdblfloat#1[#2]{%
266     \xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}

                Moved to ltoutput 93/12/16
267 %\newcount\c@topnumber
268 %\newcount\c@dbltopnumber
269 %\newcount\c@bottomnumber
270 %\newcount\c@totalnumber

\@floatplacement An analysis of \@floatplacement:
                    This should be called whenever \@colht has been set.
271 \def\@floatplacement{\global\@topnum\c@topnumber
272     % Textpage bit, global:
273     \global\@toproom \topfraction\@colht
274     \global\@botnum \c@bottomnumber
275     \global\@botroom \bottomfraction\@colht
276     \global\@colnum \c@totalnumber
277     % Floatpage bit, local:
278     \@fpmin \floatpagefraction\@colht}
279 \</2ekernel>

\@dblfloatplacement This should be called only within a group. Now changed to provide extra checks
                    in \@addtodblcol, needed when processing a BANG float.
280 \<latexrelease>\IncludeInRelease{2015/01/01}%
281 \<latexrelease> \{\@dblfloatplacement\}{float order in 2-column}%
282 \<*2ekernel | latexrelease>

```

```

283 \def\@dblfloatplacement{\global\@dbltopnum\cdbltopnumber
284 \global\@dbltoproom \dbltopfraction\@colht
285 \@textmin \@colht
286 \advance \@textmin -\@dbltoproom
287 \@fpmmin \dblfloatpagefraction\textheight
288 \@fptop \dblfpptop
289 \@fpsep \dblfpsep
290 \@fpbot \dblfpbot

\floatdepth is used in \@testwrongwidth to look for either column or
column floats. A value of 1sp signals the latter. Because of this setting
\@dblfloatplacement needs to be called inside a group which is a question
design.

291 \def\floatdepth{1sp}}%

292 </2kernel | latexrelease>
293 <latexrelease>\EndIncludeInRelease
294 <latexrelease>\IncludeInRelease{0000/00/00}%
295 <latexrelease> \floatplacement{\float order in 2-column}%
296 <latexrelease>\def \dblfloatplacement {}

```

```

297 \global \@dbltopnum \c@dbltopnumber
298 \global \@dbltoproom \dbltopfraction\@colht

```

```

299 \latexrelease \textmin \colht
300 \latexrelease \advance \textmin -\dbltoproom

```

```

301 \latexrelease) \@fpmmin \dblfloatpagefraction\textheight
302 \latexrelease) \@fptop \@dblftop
303 \latexrelease) \@fpsep \@dblpsep
304 \latexrelease) \@fpbot \@dblfpbot
305 \latexrelease)}}%
306 \latexrelease)\EndIncludeInRelease
307 \*2ekernel)

```

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.

File I: ltfloat.dtx Date: 2015/02/21 Version v1.2c 463

```

\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 \@mparbottom :=G 0
                     @reversemargin :=G true
                     END

\normalmarginpar == BEGIN \@mparbottom :=G 0
                      @reversemargin :=G false
                      END

```

End of historical L^AT_EX 2.09 comments.

`\marginpar`

```

308 \def\marginpar{%
309   \ifhmode
310     \bsphack
311     \@floatpenalty -\@Mii
312   \else
313     \@floatpenalty-\@Miii
314   \fi
315   \ifinner
316     \@parmoderr
317     \@floatpenalty\z@
318   \else
319     \@next\@currbox\@freelist{}\{}%
320     \@next\@marbox\@freelist{\global\count\@marbox\m@ne}%
321     {\@floatpenalty\z@
322       \fltovf\def\@currbox{\@tempboxa}\def\@marbox{\@tempboxa}}%
323   \fi
324   \@ifnextchar [\@xmpar\@ympar}

```

`\@xmpar`

```

325 \long\def\@xmpar[#1]#2{%
326   \@savemarbox\@marbox{#1}%
327   \@savemarbox\@currbox{#2}%
328   \@xympar}

```

`\@ympar`

```

329 \long\def\@ympar#1{%
330   \@savemarbox\@marbox{#1}%
331   \global\setbox\@currbox\copy\@marbox
332   \@xympar}

```

`\@savemarbox`

```

333 \long\def \@savemarbox #1#2{%
334   \global\setbox #1%
335     \color@vbox
336     \vtop{%
337       \hsize\marginparwidth
338       \@parboxrestore
339       \@marginparreset
340       #2%
341       \@minipagefalse
342       \outer@nobreak
343     }%
344   \color@endbox
345 }

```

`\@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@nobreake`; otherwise this command will be redundant.

```

346 \def \@marginparreset {%
347     \reset@font
348     \normalsize
349 %     \let@if@nobreake\iffalse
350 %     \let@if@noskipsec\iffalse
351 %     \@setnobreake
352     \@setminipage
353 }
```

`\@xympar`

Setting the box here is done only because the code uses `\end@float`; it will be empty and gets discarded.

```

354 \def \@xympar{%
355     \ifnum\@floatpenalty <\z@\@cons\@currlist\@marbox\fi
356     \setbox\@tempboxa
357     \color@vbox
358     \vbox \bgroup
359     \end@float
360     \@ignorefalse
361     \@esphack
362 }
```

`\reversemarginpar`

`\normalmarginpar`

```

363 \def\reversemarginpar{\global\@mparbottom\z@ \@reversemargintrue}
364 \def\normalmarginpar{\global\@mparbottom\z@ \@reversemarginfalse}

365 \message{footnotes,}
```

72.2 Footnotes

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

`\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
`\@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

```

End of historical L^AT_EX 2.09 comments.

```

\footins LATEX does use the same insert for footnotes as PLAIN.
366 \newinsert\footins
      LATEX leaves these initializations for the \footins insert.
367 \skip\footins=\bigskipamount % space added when footnote is present
368 \count\footins=1000 % footnote magnification factor (1 to 1)
369 \dimen\footins=8in % maximum footnotes per page

\footnoterule LATEX keeps PLAIN TEX's \footnoterule as the default.
370 \def\footnoterule{\kern-3\p@
371   \hrule \@width 2in \kern 2.6\p@} % the \hrule is .4pt high

\thefootnote
372 \@definecounter{footnote}
373 \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.
374 \@definecounter{mpfootnote}
375 \def\thempfootnote{{\itshape\@alph\c@mpfootnote}}

\@makefnmark Default definition.
376 %\def\@makefnmark{\hbox{\$^{\@thefnmark}\m@th$}}
377 \def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}

```


`\textsuperscript` This command provides superscript characters in the current text font. It's implementation might change!!!

```

378 \DeclareRobustCommand*\textsuperscript[1]{%
379   \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.

```

380 \def\@textsuperscript#1{%
381   {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}}

```

`\textsubscript`

```

382 \if2kernel
383 \ifx\latexrelease\IncludeInRelease{2015/01/01}%
384 \ifx\latexrelease\textsubscript{\textsubscript}%
385 \ifx\latexrelease\*2kernel
386 \DeclareRobustCommand*\textsubscript[1]{%
387   \textsubscript{\selectfont#1}}%

```

`\@textsubscript`

```

388 \def\@textsubscript#1{%
389   {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}}
390 \if2kernel\ifx\latexrelease
391 \ifx\latexrelease\EndIncludeInRelease
392 \ifx\latexrelease\IncludeInRelease{0000/00/00}%
393 \ifx\latexrelease\textsubscript{\textsubscript}%
394 \ifx\latexrelease\let\textsubscript\@undefined
395 \ifx\latexrelease\let\@textsubscript\@undefined
396 \ifx\latexrelease\EndIncludeInRelease
397 \ifx\latexrelease\*2kernel

```

`\footnotesep`

```

398 \newdimen\footnotesep

```

`\footnote`

```

399 \def\footnote{\ifnextchar[\@xfootnote{\stepcounter\@mpfn
400   \protected@xdef\@thefnmark{\thempfn}%
401   \@footnotemark\@footnotetext}}

```

`\@xfootnote`

```

402 \def\@xfootnote[#1]{%
403   \begingroup
404     \csname c@\@mpfn\endcsname #1\relax
405     \unrestored@protected@xdef\@thefnmark{\thempfn}%
406   \endgroup
407   \@footnotemark\@footnotetext}

```

`\@footnotetext`

```

408 \long\def\@footnotetext#1{\insert\footins{%
409   \reset@font\footnotesize
410   \interlinepenalty\interfootnotelinepenalty

```

```

411 \splittopskip\footnotesep
412 \splitmaxdepth \dp\strutbox \floatingpenalty \MM
413 \hsize\columnwidth \parboxrestore
414 \protected@edef\@currentlabel{%
415 \csname p@footnote\endcsname\@thefnmark
416 }%
417 \color@begingroup
418 \makefnmark{%
419 \rule\z@{\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
420 \color@endgroup}}%

\footnotemark

421 \def\footnotemark{%
422 \ifnextchar[\@xfootnotemark
423 {\stepcounter{footnote}%
424 \protected@xdef\@thefnmark{\thefootnote}%
425 \@footnotemark}}

\@xfootnotemark

426 \def\@xfootnotemark[#1]{%
427 \begingroup
428 \c@footnote #1\relax
429 \unrestored@protected@xdef\@thefnmark{\thefootnote}%
430 \endgroup
431 \@footnotemark}

\@footnotemark

432 \def\@footnotemark{%
433 \leavevmode
434 \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
435 \makefnmark
436 \ifhmode\spacefactor\@x@sf\fi
437 \relax}

\footnotetext

438 \def\footnotetext{%
439 \ifnextchar [\@xfootnotenext
440 {\protected@xdef\@thefnmark{\thempfn}%
441 \@footnotetext}}

\@xfootnotenext

442 \def\@xfootnotenext[#1]{%
443 \begingroup
444 \csname c@\@mpfn\endcsname #1\relax
445 \unrestored@protected@xdef\@thefnmark{\thempfn}%
446 \endgroup
447 \@footnotetext}

\thempfn
\@mpfn
448 \def\@mpfn{footnote}
449 \def\thempfn{\thefootnote}
450 /2ekernel)

```

File J

ltidxglo.dtx

73 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.
Historical LATEX 2.09 comments (not necessarily accurate any more):
\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.
End of historical LATEX 2.09 comments.
1 (*2ekernel)
2 \message{index,}
```

\makeindex

```
3 \def\makeindex{%
4   \newwrite\@indexfile
5   \immediate\openout\@indexfile=\jobname.idx
6   \def\index{\@bsphack\begingroup
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\begingroup \@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\begingroup
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\begingroup\@sanitize\@index}
36 </2ekernel>
```

File K

ltbibl.dtx

74 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,}
```

Historical \TeX 2.09 comments (not necessarily accurate any more):
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 NOTE argument
	is present. The default definition is : <code>\@cite{LABELS}{NOTE} ==</code> <code>BEGIN [LABELS</code> <code>IF @tempswa = T THEN , NOTE FI</code> <code>]</code>

END

`\@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`.

End of historical L^AT_EX 2.09 comments.

`\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 [{\@tempswatrue\@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{\zap@space#1 \@empty}}%
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^AT_EX 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^AT_EX, missing all citations done with `\nocite`. Thus we do only generate an error message and leave the fix for a L^AT_EX 2_ε 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@files\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 \LaTeX in the first place).

```

51 \expandafter\let\csname b@*\endcsname\@empty

```

74.1 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 L

ltpage.dtx

75 Page styles and related commands

75.1 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.

75.2 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.)

75.3 marking conventions

L^AT_EX extends T_EX'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_EX's `\botmark`.

`\rightmark` : Used in the output routine, gets the current 'right' mark. Works like T_EX'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

18 \</2kernel>
19 \*2kernel | latexrelease>
20 \<latexrelease>\IncludeInRelease{2019/10/01}%
21 \<latexrelease> \markboth{Make commands robust}%
```

`\markboth` User commands for setting L^AT_EX 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`

```

22 \DeclareRobustCommand\markboth[2]{%
23   \begingroup
24     \let\label\relax \let\index\relax \let\glossary\relax
25     \unrestored@protected@xdef\@themark {{#1}{#2}}%
26     \temptokena \expandafter{\@themark}%
27     \mark{\the\@temptokena}%
28   \endgroup
29   \if@nobreak\ifvmode\nobreak\fi\fi}

30 \DeclareRobustCommand\markright[1]{%
31   \begingroup
32     \let\label\relax \let\index\relax \let\glossary\relax

Protection is handled inside \@markright.

33   \expandafter\@markright\@themark {#1}%
34   \temptokena \expandafter{\@themark}%
35   \mark{\the\@temptokena}%
36 \endgroup
37 \if@nobreak\ifvmode\nobreak\fi\fi}

38 \</2kernel | latexrelease>
39 \<latexrelease>\EndIncludeInRelease
40 \<latexrelease>\IncludeInRelease{0000/00/00}%
```

```

41 \latexrelease)                {\markboth}{Make commands robust}%
42 \latexrelease)
43 \latexrelease)\kernel@make@fragile\markboth
44 \latexrelease)\kernel@make@fragile\markright
45 \latexrelease)
46 \latexrelease)\EndIncludeInRelease
47 \<*2ekernel)

\@markright
\leftmark 48 \def\@markright#1#2#3{\@temptokena {#1}%
\rightmark 49 \unrestored@protected@xdef\@themark{\the\@temptokena}{#3}}
50 \def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
51 \def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}

\@themark Initialise LATEX's marks without setting a TEX mark <whatsit>.
52 \def\@themark{}{}

\mark Test versions of LATEX 2ε initialised TEX'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.)

53 \DeclareRobustCommand\raggedbottom{%
54 \def\@textbottom{\vskip \z@ \@plus.0001fil}\let\@texttop\relax}

\flushbottom \flushbottom: Inverse of \raggedbottom — makes all pages the same height.

55 \DeclareRobustCommand\flushbottom{%
56 \let\@textbottom\relax \let\@texttop\relax}

\sloppy \sloppy will never (well, hardly ever) produce overfull boxes, but may produce
underfull ones. (14 June 85)

57 \DeclareRobustCommand\sloppy{%
58 \tolerance 9999%
59 \emergencystretch 3em%
60 \hfuzz .5\p@
61 \vfuzz\hfuzz}

sloppypar A sloppypar environment is equivalent to {\par \sloppy ... \par}.

62 \def\sloppypar{\par\sloppy}
63 \def\endsloppypar{\par}

\fussy Resets TEX's parameters to their normal finicky values.

64 \DeclareRobustCommand\fussy{%
65 \emergencystretch\z@
66 \tolerance 200%
67 \hfuzz .1\p@
68 \vfuzz\hfuzz}

\overfullrule LATEX default is no overfull box rule. Changed by document class option.

69 \overfullrule 0pt

70 \</2ekernel)

```

File M

ltoutput.dtx

76 Output Routine

76.1 Floats

The ‘2ekernel’ code ensures that a `\usepackage{autoout1}` is essentially ignored if a ‘full’ format is being used that has the autoloader file mode already in the format.

```
1 \def\begingroup
2 \def\makeatletter
3 \def\nfss@catcodes
4 (2ekernel)\expandafter\let\csname ver@autoout1.sty\endcsname\fmtversion
```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

```
5 (*2ekernel)
6 \message{output,}

*****
*                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: `\vadjust{\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: $\log_2 [(\text{\count I})/32]$ 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

`\NUM = [(\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.
End of historical L^AT_EX 2.09 comments.

```

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}}
12 \def\@testfalse{\global\let@if@test\iffalse}
13 \def\@testtrue {\global\let@if@test\iftrue}
14 \@testfalse
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.

```

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:

23 \ifx\@xkernel
24 \ifx\@xrelease\@xincludeinrelease{2015/10/01}%
25 \ifx\@xrelease\@xextendedfloatlist{\@xZZ}{Extended float list}%
26 \ifx\@xkernel\@xrelease
27 \let\@xelt\@xnewinsert
28 \ifx\@xkernel
29 \def\@xfreelist{%
30   \@xelt\@xA\@xelt\@xB\@xelt\@xC\@xelt\@xD\@xelt\@xE
31   \@xelt\@xF\@xelt\@xG\@xelt\@xH\@xelt\@xI\@xelt\@xJ
32   \@xelt\@xK\@xelt\@xL\@xelt\@xM\@xelt\@xN
33   \@xelt\@xO\@xelt\@xP\@xelt\@xQ\@xelt\@xR}
34 \@xfreelist
35 \ifx\@xkernel
36 \ifx\@xnumexpr\@xundefined\else
37 \def\@xreserved@a{%
38   \@xelt\@xS\@xelt\@xT\@xelt\@xU\@xelt\@xV
39   \@xelt\@xW\@xelt\@xX\@xelt\@xY\@xelt\@xZ
40   \@xelt\@xAA\@xelt\@xBB\@xelt\@xCC\@xelt\@xDD\@xelt\@xEE
41   \@xelt\@xFF\@xelt\@xGG\@xelt\@xHH\@xelt\@xII\@xelt\@xJJ
42   \@xelt\@xKK\@xelt\@xLL\@xelt\@xMM\@xelt\@xNN
43   \@xelt\@xOO\@xelt\@xPP\@xelt\@xQQ\@xelt\@xRR
44   \@xelt\@xSS\@xelt\@xTT\@xelt\@xUU\@xelt\@xVV
45   \@xelt\@xWW\@xelt\@xXX\@xelt\@xYY\@xelt\@xZZ}
46 \@xreserved@a
47 \def\@xelt{\@xnoexpand\@xelt\@xnoexpand}
48 \edef\@xfreelist{\@xfreelist\@xreserved@a}
49 \fi
50 \let\@xreserved@a\relax
51 \let\@xelt\relax
52 \ifx\@xkernel\@xrelease
53 \ifx\@xrelease\@xendincludeinrelease
54 \ifx\@xrelease\@xincludeinrelease{0000/00/00}%
55 \ifx\@xrelease\@xextendedfloatlist{\@xZZ}{Extended float list}%
56 \ifx\@xrelease\@xdef\@xfreelist{%
57 \ifx\@xrelease\@xelt\@xA\@xelt\@xB\@xelt\@xC\@xelt\@xD\@xelt\@xE
58 \ifx\@xrelease\@xelt\@xF\@xelt\@xG\@xelt\@xH\@xelt\@xI\@xelt\@xJ
59 \ifx\@xrelease\@xelt\@xK\@xelt\@xL\@xelt\@xM\@xelt\@xN
60 \ifx\@xrelease\@xelt\@xO\@xelt\@xP\@xelt\@xQ\@xelt\@xR}
61 \ifx\@xrelease\@xinscunt=234
62 \ifx\@xrelease\@xendincludeinrelease
63 \ifx\@xkernel
64 \gdef\@xtoplist{}
65 \gdef\@xbotlist{}
66 \gdef\@xmidlist{}
67 \gdef\@xcurrelist{}

```

```

68 \gdef\@deferlist{}
69 \gdef\@dbltoplist{}

```

The new algorithm stores page wide floats together with column floats in a single `\@deferlist` list. We keep `\@dbldeferlist` initialised as empty so that packages that are testing for deferred floats can use the same code for old or new float handling.

```

70 \gdef\@dbldeferlist{}

PAGE LAYOUT PARAMETERS

71 \newdimen\topmargin
72 \newdimen\oddsidemargin
73 \newdimen\evensidemargin
74 \let\@themargin=\oddsidemargin
75 \newdimen\headheight
76 \newdimen\headsep
77 \newdimen\footskip
78 \newdimen\textheight
79 \newdimen\textwidth
80 \newdimen\columnwidth
81 \newdimen\columnsep
82 \newdimen\columnseprule
83 \newdimen\marginparwidth
84 \newdimen\marginparsep
85 \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.

```

86 \newbox\@begindvibox
87 \DeclareRobustCommand \AtBeginDvi [1]{%
88   \global \setbox \@begindvibox
89     \vbox{\unvbox \@begindvibox #1}%
90 }

```

`\@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?

```

91 \newdimen\@maxdepth
92 \@maxdepth = \maxdepth

```

`\paperheight` New `\paper...` registers.

```

\paperwidth 93 \newdimen\paperheight
94 \newdimen\paperwidth

```

`\if@insert` Local switches first:

```

\if@fcolmade 95 \newif \if@insert

```

`\if@specialpage` These should definitely be global:

```

\if@firstcolumn 96 \newif \if@fcolmade
\if@twocolumn 97 \newif \if@specialpage \@specialpagefalse
\if@twoside

```

```

\if@reversemarginpar
\if@mparswitch
\col@number

```


These should be global but are not always set globally in other files.

```
98 \newif \if@firstcolumn \@firstcolumntrue
99 \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?

```
100 \newif \if@twoside \@twosidefalse
101 \newif \if@reversemargin \@reversemarginfalse
102 \newif \if@mparswitch \@mparswitchfalse
```

This counter has been imported from ‘multicol’.

```
103 \newcount \col@number
104 \col@number \@ne
```

Historical L^AT_EX 2.09 comments (not necessarily accurate any more):

INTERNAL REGISTERS

```
105 \newcount\@topnum
106 \newdimen\@toproom
107 \newcount\@dbltopnum
108 \newdimen\@dbltoproom
109 \newcount\@botnum
110 \newdimen\@botroom
111 \newcount\@colnum
112 \newdimen\@textmin
113 \newdimen\@fpmin
114 \newdimen\@colht
115 \newdimen\@colroom
116 \newdimen\@pageht
117 \newdimen\@pagedp
118 \newdimen\@mparbottom \@mparbottom\z@
119 \newcount\@currtype
120 \newbox\@outputbox
121 \newbox\@leftcolumn
122 \newbox\@holdpg

123 \def\@thehead{\@oddhead} % initialization
124 \def\@thefoot{\@oddfoot}
```

End of historical L^AT_EX 2.09 comments.

`\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.

```
125 \def\clearpage{%
126   \ifvmode
127     \ifnum \@dbltopnum =\m@ne
128       \ifdim \pagetotal <\topskip
129         \hbox{}%
130       \fi
131     \fi
132   \fi
```

```

133 \newpage
134 \write\m@ne{}%
135 \vbox{}%
136 \penalty -\@Mi
137 }

\cleardoublepage

138 \def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
139 \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
140 \endkernel)

\onecolumn

141 (*2ekernel | ftrace)
142 \def\onecolumn{%
143 \clearpage
144 \global\columnwidth\textwidth
145 \global\hsize\columnwidth
146 \global\linewidth\columnwidth
147 \global\@twocolumnfalse
148 \col@number \@ne
149 \@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.

150 \endkernel | ftrace)
151 (latexrelease)\IncludeInRelease{2017/04/15}%
152 (latexrelease) {\newpage}{Check depth of page}%
153 (*2ekernel | latexrelease | ftrace)
154 \def \newpage {%
155 \if@noskipsec
156 \ifx \@nodocument\relax
157 \leavevmode
158 \global \@noskipsecfalse
159 \fi
160 \fi
161 \if@inlabel
162 \leavevmode
163 \global \@inlabelfalse
164 \fi
165 \if@nobreak \@nobreakfalse \everypar{}\fi
166 \par

The \vfil at the end of the macro before the break penalty will normally result
in the page being run short, even with \flushbottom in effect (in contrast to the
behavior of \pagebreak). However, if there is some explicit stretch on the page,
say, a \vfill, it has the undesired side-effect, that the last line will not align
at its baseline if it contains characters going below the baseline, as the value of
\prevdepth is no longer taken into account by TEX. So we back up by that amount
(or by \maxdepth if it is really huge), to mimic the normal behavior without the
\newpage.

```

```

167 \ifdim\prevdepth>\z@
168 \vskip -%
169 \ifdim\prevdepth>\maxdepth
170 \maxdepth
171 \else
172 \prevdepth
173 \fi
174 \fi
175 \vfil
176 \penalty -\@M}
177 /2ekernel| latexrelease| fltrace)
178 (latexrelease)\EndIncludeInRelease
179 (latexrelease)\IncludeInRelease{0000/00/00}%
180 (latexrelease) {\newpage}{Check depth of page}%
181 (latexrelease)\def \newpage {%
182 (latexrelease) \if@noskipsec
183 (latexrelease) \ifx \@nodocument\relax
184 (latexrelease) \leavevmode
185 (latexrelease) \global \@noskipsecfalse
186 (latexrelease) \fi
187 (latexrelease) \fi
188 (latexrelease) \if@inlabel
189 (latexrelease) \leavevmode
190 (latexrelease) \global \@inlabelfalse
191 (latexrelease) \fi
192 (latexrelease) \if@nobreak \@nobreakfalse \everypar{}\fi
193 (latexrelease) \par
194 (latexrelease) \vfil
195 (latexrelease) \penalty -\@M}
196 (latexrelease)\EndIncludeInRelease
197 (*2ekernel| fltrace)

```

\@emptycol It may be better to use an invisible rule rather than an empty box here.

```
198 \def \@emptycol {\vbox{}\penalty -\@M}
```

\twocolumn There are several bug fixes to the two-column stuff here.

```

\@topnewpage 199 \def \twocolumn {%
200 \clearpage
201 \global\columnwidth\textwidth
202 \global\advance\columnwidth-\columnsep
203 \global\divide\columnwidth\tw@
204 \global\hsize\columnwidth
205 \global\linewidth\columnwidth
206 \global\@twocolumntrue
207 \global\@firstcolumntrue
208 \col@number \tw@

```

There is no reason to put a \@dblfloatplacement here since \@topnewpage ignores these settings. The \@floatplacement is needed in case this comes after some changes.

```

209 \@ifnextchar [\@topnewpage\@floatplacement
210 }

```

Note that here, getting a box from the freelist can assume success since this comes just after a \clearpage.

```

211 \long\def \@topnewpage [#1]{%
212   \@nodocument
213   \@next\@currbox\@freelist{}-{}%
214   \global \setbox\@currbox
215     \color@vbox
216     \normalcolor
217     \vbox {%
218       \hsize\textwidth
219       \@parboxrestore
220       \col@number \@ne
221       #1%
222       \vskip -\dbltextfloatsep
223     }%
224   \color@endbox

```

Added size test and warning message; perhaps we should use an error message.

```

225   \ifdim \ht\@currbox>\textheight
226     \ht\@currbox \textheight
227   \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.

```

228   \global \count\@currbox \tw@
229   \@tempdima -\ht\@currbox
230   \advance \@tempdima -\dbltextfloatsep
231   \global \advance \@colht \@tempdima
232   \ifx \@dbltoplist \@empty
233   \else
234     \@latexerr{Float(s) lost}\@ehb
235     \let \@dbltoplist \@empty
236   \fi
237   \@cons \@dbltoplist \@currbox

```

This setting of `\@dbltopnum` is used only to change the typesetting in `\@combinedblfloats`.

```

238   \global \@dbltopnum \m@ne
239   (*trace)
240     \fl@trace{dbltopnum set to -1 (= \the \@dbltopnum) (topnewpage)}%
241   (/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.

```

242 \ifdim \@colht<2.5\baselineskip
243 \latex@warning@no@line {Optional argument of \noexpand\twocolumn
244 too tall on page \thepage}%
245 \emptycol
246 \if@firstcolumn
247 \else
248 \emptycol
249 \fi
250 \else
251 \global \vsize \@colht
252 \global \@colroom \@colht
253 \floatplacement
254 \fi
255 }
```

`\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.

```

256 \output {%
257 \let \par \@par
258 \ifnum \outputpenalty<-\@M
259 \specialoutput
260 \else
261 \makecol
262 \opcol
```

Moved to `\@opcol: \floatplacement`.

```

263 \@startcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startcolumn`.

```

264 \@whilesw \if@fcolmade \fi
265 {%
266 (*trace)
267 \fl@trace{PAGE: float \if@twocolumn column \else page \fi
268 completed}%
269 (/trace)
270 \@opcol\@startcolumn}%
271 \fi
272 \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`.)

```

273 \ifdim \@colroom<1.5\baselineskip
274 \ifdim \@colroom<\textheight
275 \latex@warning@no@line {Text page \thepage\space
276 contains only floats}%
277 \@emptycol
278 % \if@twocolumn
279 % \if@firstcolumn
280 % \else
281 % \@emptycol
282 % \fi
283 % \fi
284 \else
285 \global \vsize \@colroom
286 \fi
287 \else
288 \global \vsize \@colroom
289 \fi
290 \else
291 \global \vsize \maxdimen
292 \fi
293 }

```

Historical \LaTeX 2.09 comments (not necessarily accurate any more):

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.)

End of historical \LaTeX 2.09 comments.

```

294 \gdef\@specialoutput{%
295 \ifnum \outputpenalty>-\@Mii
296 \doclearpage
297 \else
298 \ifnum \outputpenalty<-\@Miii
299 \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
300 \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
301 \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 \ccclv`.

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.

```
302      \global \setbox\@holdpg \vbox{%
303              \unvbox\@holdpg
304              \unvbox\ccclv
```

We must now remove the box added by the float mechanism and the `\topskip` glue therefore added above it by `TEX`.

```
305              \setbox\@tempboxa \lastbox
306              \unskip
307              }%
```

These two are needed as separate dimensions only by `\@addmarginpar`; for other purposes we put the whole size into `\@pageht` (see below).

```
308      \@pagedp \dp\@holdpg
309      \@pageht \ht\@holdpg
310      \unvbox \@holdpg
311      \@next\@currbox\@currlst{%
312      \ifnum \count\@currbox>\z@
```

Putting the whole size into `\@pageht` (see above).

```
313      \advance \@pageht \@pagedp
314      \ifvoid\footins \else
315      \advance \@pageht \ht\footins
316      \advance \@pageht \skip\footins
317      \advance \@pageht \dp\footins
318      \fi
319      \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?

```
320      \ifdim \wd\@kludgeins=\z@
321      \advance \@pageht \ht\@kludgeins
322      (*trace)
323      \fl@trace {Extra size added: \the \ht\@kludgeins}%
324      </trace>
325      \fi
326      \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.

```
327      \@reinserts
328      \@addtocurcol
329      \else
330      \@reinserts
331      \@addmarginpar
```

```

332         \fi
333     }\@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.

```

334         \ifnum \outputpenalty<\z@
335             \if@nobreak
336                 \nobreak
337             \else
338                 \addpenalty \interlinepenalty
339             \fi
340         \fi
341     \fi
342 \fi
343 }
344 \if2kernel | 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).

```

345 \<latexrelease>\IncludeInRelease{2015/01/01}%
346 \<latexrelease>         {\@testwrongwidth}{float order in 2-column}%
347 \if2kernel | latexrelease | fltrace)

348 \def\@testwrongwidth #1{%
349     \ifdim\dp#1=\f@depth
350 \<*trace>
351     \fl@trace{\string#1
352         \ifdim\f@depth=\z@ single \else double \fi
353         column float -- ok}%
354 \</trace>
355     \else
356         \global\@testtrue
357 \<*trace>
358     \fl@trace{\string#1
359         \ifdim\f@depth=\z@ double \else single \fi
360         column float -- wrong}%
361 \</trace>
362     \fi}%

```

Normally looking for single column floats, which have zero depth.

```

363 \let\f@depth\z@
364 \if2kernel | latexrelease | fltrace)
365 \<latexrelease>\EndIncludeInRelease
366 \<latexrelease>\IncludeInRelease{0000/00/00}%
367 \<latexrelease>         {\@testwrongwidth}{float order in 2-column}%
368 \<latexrelease>\let\@testwrongwidth\@undefined
369 \<latexrelease>\let\f@depth\@undefined
370 \<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 uncancelable.

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.

```

371 \latexrelease\IncludeInRelease{2015/01/01}{\@doclearpage}%
372 \latexrelease\float order in 2-column}%
373 \*2ekernel|latexrelease
374 \def \@doclearpage {%
375     \ifvoid\footins

376     \ifvbox\@kludgeins
377     {\setbox \@tempboxa \box \@kludgeins}%
378 \*trace)
379     \fl@trace {kludgeins box made void}%
380 \*trace)
381     \fi
382     \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
383     \setbox\@tempboxa\box\@cclv
384     \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
385     \global \let \@toplist \@empty
386     \global \let \@botlist \@empty
387     \global \@colroom \@colht
388     \ifx \@currlist\@empty
389     \else
390     \@latexerr{Float(s) lost}\@ehb
391     \global \let \@currlist \@empty
392     \fi
393     \@makefcolumn\@deferlist
394     \@whilesw\if@colmade \fi{\@opcol\@makefcolumn\@deferlist}%
395     \if@twocolumn
396     \if@firstcolumn

397     \xdef\@deferlist{\@dbltoplist\@deferlist}%
398     \global \let \@dbltoplist \@empty
399     \global \@colht \textheight
400     \begingroup
401     \@dblfloatplacement
402     \@makefcolumn\@deferlist
403     \@whilesw\if@colmade \fi{\@outputpage
404     \@makefcolumn\@deferlist}%

405     \endgroup
406     \else
407     \vbox{}\clearpage
408     \fi
409     \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.

```

410 \ifx\@deferlist\@empty \else\clearpage \fi
411 \else
412 \setbox\@cclv\vbox{\box\@cclv\vfil}%
413 \@makecol\@opcol
414 \clearpage
415 \fi
416 }%
417 //2kernel | latexrelease>
418 <latexrelease>\EndIncludeInRelease
419 <latexrelease>\IncludeInRelease{0000/00/00}{\@docclearpage}%
420 <latexrelease> {float order in 2-column}%
421 <latexrelease>\def \@docclearpage {%
422 <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.

```

423 <latexrelease> \ifvbox\@kludgeins
424 <latexrelease> {\setbox \@tempboxa \box \@kludgeins}%
425 <*trace>
426 <latexrelease> \fl@trace {kludgeins box made void}%
427 </trace>
428 <latexrelease> \fi
429 <latexrelease> \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
430 <latexrelease> \setbox\@tempboxa\box\@cclv
431 <latexrelease> \xdef\@deferlist{\@toplist\@botlist\@deferlist}%

432 <latexrelease> \global \let \@toplist \@empty
433 <latexrelease> \global \let \@botlist \@empty
434 <latexrelease> \global \colroom \colht
435 <latexrelease> \ifx \@currlist\@empty
436 <latexrelease> \else
437 <latexrelease> \@latexerr{Float(s) lost}\@ehb

438 <latexrelease> \global \let \@currlist \@empty
439 <latexrelease> \fi
440 <latexrelease> \@makefcolumn\@deferlist
441 <latexrelease> \@whiles\if@fcolmade \fi
442 <latexrelease> {\@opcol\@makefcolumn\@deferlist}%
443 <latexrelease> \if@twocolumn
444 <latexrelease> \if@firstcolumn
445 <latexrelease> \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%

446 <latexrelease> \global \let \@dbltoplist \@empty
447 <latexrelease> \global \@colht \textheight
448 <latexrelease> \begingroup
449 <latexrelease> \@dblfloatplacement
450 <latexrelease> \@makefcolumn\@dbldeferlist
451 <latexrelease> \@whiles\if@fcolmade \fi
452 <latexrelease> {\@outputpage\@makefcolumn\@dbldeferlist}%
453 <latexrelease> \endgroup
454 <latexrelease> \else
455 <latexrelease> \vbox{}\clearpage
456 <latexrelease> \fi
457 <latexrelease> \fi

```

```

458 \latexrelease\else
459 \latexrelease\setbox\@cclv\vbox{\box\@cclv\vfil}%
460 \latexrelease\@makecol\@opcol
461 \latexrelease\clearpage
462 \latexrelease\fi
463 \latexrelease}%
464 \latexrelease\EndIncludeInRelease

```

`\@opcol` Several changes in detail here.

```

465 (*2ekernel|fltrace)
466 \def \@opcol {%
467   \if@twocolumn
468     \@outputdblcol
469   \else
470     \@outputpage
471 (*trace)
472   \fl@trace{PAGE: one column (float? see above) page completed}%
473 /trace)

```

Not needed since it comes after `\@outputpage`:

```

474 %   \global\@colht\textheight
475 \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.

```

476 \global \mparbottom \z@ \global \textfloatsheight \z@
477 \floatplacement
478 }
479 /2ekernel|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.

```

480 (*2ekernel)
481 \gdef \@makecol {%
482   \ifvoid\footins
483     \setbox\@outputbox \box\@cclv
484   \else
485     \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.

```

486       \boxmaxdepth \@maxdepth

487 %       \@tempdima\dp\@cclv
488       \unvbox \@cclv
489 %       \vskip-\@tempdima
490       \vskip \skip\footins

491       \color@begingroup
492       \normalcolor

```

```

493         \footnoterule
494         \unvbox \footins
495         \color@endgroup
496     }%
497 \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.

```

498 \let\@elt\relax
499 \xdef\@freelist{\@freelist\@midlist}%

500 \global \let \@midlist \@empty
501 \@combinefloats

```

The variations start here in case `\enlargethispage` has been used.

```

502 \ifvbox\@kludgeins
503     \@makespecialcolbox
504 \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@`.

```

505     \setbox\@outputbox \vbox to\@colht {%
506 %         \boxmaxdepth \maxdepth           %??
507         \@texttop
508         \dimen@ \dp\@outputbox
509         \unvbox \@outputbox
510         \vskip -\dimen@
511         \@textbottom
512     }%
513 \fi
514 \global \maxdepth \@maxdepth
515 }

```

\@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.

```

516 \gdef \@reinserts{%
517     \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
518     \ifvbox\@kludgeins\insert\@kludgeins

```

```

519             {\unvbox\@kludgeins}\fi
520 }
521 \end{kernel}

```

\@makespecialcolbox This implements certain variations in page-makeup.

```

522 (*2kernel | fltrace)
523 \gdef \@makespecialcolbox {%
524 (*trace)
525   \fl@trace{Kludgeins ht \the\ht\@kludgeins\space
526             dp \the\dp\@kludgeins\space
527             wd \the\wd\@kludgeins}%
528 \end{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’.

```

529   \setbox\@outputbox \vbox {%
530     \texttop
531     \dimen@ \dp\@outputbox
532     \unvbox\@outputbox
533     \vskip-\dimen@
534   }%
535   \@tempdima \@colht
536   \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 T_EX level!).

This needs T_EX3 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.

Their 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.

```

537   \advance \@tempdima -\ht\@outputbox
538   \advance \@tempdima \pageshrink
539 (*trace)
540   \fl@trace {Natural ht of col: \the \ht\@outputbox}%
541   \fl@trace {\string \@colht: \the \@colht}%
542   \fl@trace {Pageshrink added: \the \pageshrink}%
543   \fl@trace {Hence, space added: \the \@tempdima}%
544 \end{trace}
545   \setbox\@outputbox \vbox to \@colht {%
546 %       \boxmaxdepth \maxdepth

```

```

547      \unvbox\@outputbox
548      \vskip \@tempdima
549      \@textbottom
550      }%

```

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.

```

551      \else
552      \advance \@tempdima -\ht\@kludgeins
553      (*trace)
554      \fl@trace {Natural ht of col: \the \ht\@outputbox}%
555      \fl@trace {\string \@colht: \the \@colht}%
556      \fl@trace {Extra size added: -\the \ht \@kludgeins}%
557      \fl@trace {Hence, height of inner box: \the \@tempdima}%
558      \fl@trace {Max? pageshrink available: \the \pageshrink}%
559      (/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.

```

560      \setbox \@outputbox \vbox to \@colht {%
561      \vbox to \@tempdima {%
562      \unvbox\@outputbox
563      \@textbottom}%
564      \vss}%
565      \fi

```

Finally we need to explicitly make the insert box void.

```

566      {\setbox \@tempboxa \box \@kludgeins}%
567      (*trace)
568      \fl@trace {kludgeins box made void}%
569      (/trace)
570      }
571      (/2ekernel | fltrace)

```

\@texttop These do nothing as a default.

```

\@textbottom 572 (*2ekernel)
573 \let \@texttop \relax
574 \let \@textbottom \relax

```

\@resetactivechars RmS 93/09/06: added hook to protect against certain active characters in the output routine. Default checks are for active space and end-of-line.

```

\@activechar@info
575 \def\@activechar@info #1{%
576     \@latex@info@no@line {Active #1 character found while
577                           output routine is active
578                           \MessageBreak
579                           This may be a bug in a package file
580                           you are using}%
581 }

```

Do not put any spaces in this next bit!

```
582 \begingroup
583 \obeylines\obeyspaces%
584 \catcode'\'\active%
585 \gdef\@resetactivechars{%
586 \def^^M{\@activechar@info{EOL}\space}%
587 \def {\@activechar@info{space}\space}%
588 \let'\active@math@prime}%
589 \endgroup
```

`\@outputpage` The `\color@hbox` hooks here are used to avoid putting just a colour special into an otherwise empty box (in a header or footer). These boxes are often set to be completely empty and so adding a special produces a very underfull box message.

`\@shipoutsetup` There has been extensive tidying up of the old code here; including the removal of a level of grouping.

`\@writsetup` 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 to prevent mis-use of internal commands as hooks.

```
590 \</2ekernel>
591 \<latexrelease>\IncludeInRelease{2017/04/15}%
592 \<latexrelease> {\@outputpage}{Reset language for hyphenation}%
593 \<*2ekernel|latexrelease>
594 \def\@outputpage{%
```

The `\endgroup` is put in by `\aftergroup`.

```
595 \begingroup
```

Now all the set-up stuff has been in-lined for Frank.

First the stuff for the writes.

From here ... was in the command `\@writsetup`.

```
596 \let \protect \noexpand
```

RmS 93/08/19: Redefined accents to allow changes in font encoding; but exactly why was this needed?

Reset `\language` to the value current at `\begin{document}`. In particular this ensures that a pagebreak in `verbatim` does not prevent hyphenation in the page head.

```
597 \language\document@default@language
```

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?

```
598 \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.

```
599 \global\let\@@if@newlist@if@newlist
600 \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
```

```
601 \@parboxrestore
```

... to here was in the command `\@writesetup`.

```
602 \shipout \vbox{%
603   \set@typeset@protect
604   \aftergroup \endgroup
```

Correct? or just restore by ending the group?

```
605   \aftergroup \set@typeset@protect
```

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`.

```
606 \if@specialpage
607   \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
608   \fi
609 \if@twoside
610   \ifodd\count\z@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
611     \let\@themargin\oddsidemargin
612   \else \let\@thehead\@evenhead
613     \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
614   \fi
615 \fi
```


The rest was always inside the box.

RmS 91/08/15: aded this line:

```
616 \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.

```
617 \normalsize
```

Reset the space factors.

```
618 \normalsfcodes
```

Reset these here (previously reset separately for head and foot)

```
619 \let\label\@gobble
```

```
620 \let\index\@gobble
```

```
621 \let\glossary\@gobble
```

```
622 \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
```

... to here was in the command `\@shipoutsetup`.

```
623 \@beginndvi
```

```
624 \vskip \topmargin
```

```
625 \moveright\@themargin \vbox {%
```

```
626 \setbox\@tempboxa \vbox to\headheight{%
```

```
627 \vfil
```

```
628 \color@hbox
```

```
629 \normalcolor
```

```
630 \hb@xt@\textwidth{\@thehead}%
```

```
631 \color@endbox
```

22 Feb 87

```
632 }%
```

```
633 \dp\@tempboxa \z@
```

```
634 \box\@tempboxa
```

```
635 \vskip \headsep
```

```
636 \box\@outputbox
```

```
637 \baselineskip \footskip
```

```
638 \color@hbox
```

```
639 \normalcolor
```

```
640 \hb@xt@\textwidth{\@thefoot}%
```

```
641 \color@endbox
```

```
642 }%
```

```
643 }%
```

`\endgroup` now inserted by `\aftergroup`

Restore `\if@newlist`

```
644 \global\let\if@newlist\@if@newlist
```

```
645 \global \colht \textheight
```

```
646 \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.

```
647 \let\firstmark\botmark
```

```
648 }
```

```

649 </2ekernel | latexrelease>
650 <latexrelease>\EndIncludeInRelease
651 <latexrelease>\IncludeInRelease{0000/00/00}%
652 <latexrelease> {\@outputpage}{Reset language for hyphenation}%
653 <latexrelease>\def\@outputpage{%
654 <latexrelease>\begingroup
655 <latexrelease> \let \protect \noexpand
656 <latexrelease> \@resetactivechars
657 <latexrelease> \global\let\@if@newlist@if@newlist
658 <latexrelease> \global\@newlistfalse
659 <latexrelease> \@parboxrestore
660 <latexrelease> \shipout \vbox{%
661 <latexrelease> \set@typeset@protect
662 <latexrelease> \aftergroup \endgroup
663 <latexrelease> \aftergroup \set@typeset@protect
664 <latexrelease> \if@specialpage
665 <latexrelease> \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
666 <latexrelease> \fi
667 <latexrelease> \if@twoside
668 <latexrelease> \ifodd\count\z@
669 <latexrelease> \let\@thehead\@oddhead \let\@thefoot\@oddfoot
670 <latexrelease> \let\@themargin\oddsidemargin
671 <latexrelease> \else \let\@thehead\@evenhead
672 <latexrelease> \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
673 <latexrelease> \fi
674 <latexrelease> \fi
675 <latexrelease> \reset@font
676 <latexrelease> \normalsize
677 <latexrelease> \normalsfcodes
678 <latexrelease> \let\label\@gobble
679 <latexrelease> \let\index\@gobble
680 <latexrelease> \let\glossary\@gobble
681 <latexrelease> \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
682 <latexrelease> \@beginndvi
683 <latexrelease> \vskip \topmargin
684 <latexrelease> \moveright\@themargin \vbox {%
685 <latexrelease> \setbox\@tempboxa \vbox to\headheight{%
686 <latexrelease> \vfil
687 <latexrelease> \color@hbox
688 <latexrelease> \normalcolor
689 <latexrelease> \hb@xt@\textwidth{\@thehead}%
690 <latexrelease> \color@endbox
691 <latexrelease> }%
692 <latexrelease> \dp\@tempboxa \z@
693 <latexrelease> \box\@tempboxa
694 <latexrelease> \vskip \headsep
695 <latexrelease> \box\@outputbox
696 <latexrelease> \baselineskip \footskip
697 <latexrelease> \color@hbox
698 <latexrelease> \normalcolor
699 <latexrelease> \hb@xt@\textwidth{\@thefoot}%
700 <latexrelease> \color@endbox
701 <latexrelease> }%
702 <latexrelease> }%

```

```

703 <latexrelease> \global\let@if@newlist\@if@newlist
704 <latexrelease> \global \colht \textheight
705 <latexrelease> \stepcounter{page}%
706 <latexrelease> \let\firstmark\botmark
707 <latexrelease>}
708 <latexrelease>\EndIncludeInRelease
709 <*2ekernel>

```

\@begindivi 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.

```

710 \def \@begindivi{%
711   \unvbox \@begindivibox
712   \global\let \@begindivi \@empty
713 }

```

\@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 **\@cflt** it has been moved there; this also agrees with the original pseudocode.

```

714 \def \@combinefloats {%
715 %   \boxmaxdepth \maxdepth
716   \ifx \@toplist\@empty \else \@cflt \fi
717   \ifx \@botlist\@empty \else \@cflb \fi
718 }

719 \def \@cflt{%
720   \let \@elt \@comflelt
721   \setbox\@tempboxa \vbox{%
722     \@toplist
723     \setbox\@outputbox \vbox{%
724       \boxmaxdepth \maxdepth
725       \unvbox\@tempboxa
726       \vskip -\floatsep
727       \topfigrule
728       \vskip \textfloatsep
729       \unvbox\@outputbox
730     }%
731     \let\@elt\relax
732     \xdef\@freelist{\@freelist\@toplist}%
733     \global\let\@toplist\@empty
734 }

735 \def \@cflb {%
736   \let\@elt\@comflelt
737   \setbox\@tempboxa \vbox{%
738     \@botlist
739     \setbox\@outputbox \vbox{%
740       \unvbox\@outputbox
741       \vskip \textfloatsep
742       \botfigrule
743       \unvbox\@tempboxa
744       \vskip -\floatsep
745     }%
746     \let\@elt\relax

```

```

747 \xdef\@freelist{\@freelist\@botlist}%
748 \global \let \@botlist\@empty
749 }

```

```

\@comflelt
\@comdblfilelt
\@combinedblfloats
750 \def\@comflelt#1{\setbox\@tempboxa
751 \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
752 \def\@comdblfilelt#1{\setbox\@tempboxa
753 \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
754 \def \@combinedblfloats{%
755 \ifx \@dbltoplist \@empty
756 \else
757 \setbox\@tempboxa \vbox{}%
758 \let \@elt \@comdblfilelt
759 \@dbltoplist
760 \let \@elt \relax
761 \xdef \@freelist {\@freelist\@dbltoplist}%
762 \global\let \@dbltoplist \@empty
763 \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.

```

764 {\boxmaxdepth\maxdepth %% probably not needed, CAR
765 \unvbox\@tempboxa\vskip-\dblfloatsep

```

Here we need different typesetting if the top float comes from `\@topnewpage`.

```

766 \ifnum \@dbltopnum>\m@ne
767 \dblfigrule
768 \fi
769 \vskip \dbltextfloatsep

```

If pdf links are present in the galley and those links get broken across pages they have to end up being on the same level of boxing (even if not actually in the same structure) due to some engine restrictions in pdf \TeX and Lua \TeX . We therefore unbox `\@outputbox` here (which only contains a single `\hbox`) so that this case has the same boxing level as a normal twocolumn page without top floats.

```

770 \unvbox\@outputbox
771 }%
772 \fi
773 }
774 \end{kernel}

```

`\@startcolumn` We could combine (most of) these two into `\@startcol <list>`. Note that `\@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`.

```

775 \if*2kernel|ftrace)
776 \def \@startcolumn {%
777 \global \@colroom \@colht
778 \@tryfcolumn \@deferlist
779 \if@fcolmade

```

```

780 <*trace>
781   \fl@trace{PAGE: float \if@twocolumn column \else page \fi
782             completed}%
783 </trace>
784 \else

785   \begingroup
786     \let \reserved@b \@deferlist
787     \global \let \@deferlist \@empty
788     \let \@elt \@scolelt
789     \reserved@b
790   \endgroup
791 \fi
792 }

```

This one does not need to set \@colht.

```

793 </2ekernel | fltrace>
794 <latexrelease | fltrace>\IncludeInRelease{2015/01/01}%
795 <latexrelease | fltrace> {\@startdblcolumn}{float order in 2-column}%
796 <*2ekernel | latexrelease | fltrace>
797 \def \@startdblcolumn {%
798   \@tryfcolumn \@deferlist
799   \if@fcolmade
800 <fltrace>   \fl@trace{PAGE: double float page completed}%
801 \else
802   \begingroup
803     \let \reserved@b \@deferlist
804     \global \let \@deferlist \@empty
805     \let \@elt \@sdblcolelt
806     \reserved@b
807   \endgroup
808 \fi
809 }%
810 </2ekernel | latexrelease | fltrace>
811 <latexrelease | fltrace>\EndIncludeInRelease
812 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
813 <latexrelease | fltrace> {\@startdblcolumn}{float order in 2-column}%
814 <latexrelease | fltrace>\def \@startdblcolumn {%

```

Not needed since this always comes after \@outputpage:

```

815 <latexrelease | fltrace>% \global \@colht \textheight
816 <latexrelease | fltrace> \@tryfcolumn \@dbldeferlist
817 <latexrelease | fltrace> \if@fcolmade
818 <*trace>
819 <latexrelease | fltrace>   \fl@trace{PAGE: double float page completed}%
820 </trace>
821 <latexrelease | fltrace> \else

822 <latexrelease | fltrace>   \begingroup
823 <latexrelease | fltrace>     \let \reserved@b \@dbldeferlist
824 <latexrelease | fltrace>     \global \let \@dbldeferlist \@empty
825 <latexrelease | fltrace>     \let \@elt \@sdblcolelt
826 <latexrelease | fltrace>     \reserved@b
827 <latexrelease | fltrace>   \endgroup
828 <latexrelease | fltrace> \fi

```

```

829 \latexrelease | fltrace\}%
830 \latexrelease | fltrace\EndIncludeInRelease
831 \*2ekernel | fltrace)

\@tryfccolumn Now tests if its list is empty before any further exertion.

832 \def \@tryfccolumn #1{%
833   \global \@fcolmadefalse
834   \ifx #1\@empty
835     \else
836     \*trace)
837     \fl@trace{PAGE: try float \if@twocolumn column/page\else page\fi
838               ---\string #1}%
839     \fl@trace{----- \string #1: #1}%
840   \</trace)

841   \xdef\@trylist{#1}%
842   \global \let \@failedlist \@empty
843   \begingroup
844     \let \@elt \@xtryfc \@trylist
845   \endgroup
846   \if@fcolmade
847     \@vtryfc #1%
848   \fi
849 \fi
850 }
851 \</2ekernel | fltrace)

852 \*2ekernel)

\@scolelt
853 \def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}

\@sdblcolelt
854 \def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}

\@vtryfc
855 \def\@vtryfc #1{%
856   \global\setbox\@outputbox\vbox{}%
857   \let\@elt\@wtryfc
858   \@flsucceed
859   \global\setbox\@outputbox \vbox to\@colht{%
860     \vskip \@fptop
861     \vskip -\@fpsep
862     \unvbox \@outputbox
863     \vskip \@fpbot}%
864   \let\@elt\relax
865   \xdef #1{\@failedlist\@flfail}%
866   \xdef\@freelist{\@freelist\@flsucceed}}

\@wtryfc
867 \def\@wtryfc #1{%
868   \global\setbox\@outputbox\vbox{%
869     \unvbox\@outputbox
870     \vskip\@fpsep
871     \box #1}}

```

\@xtryfc

```
872 </2ekernel>
873 <latexrelease>\IncludeInRelease{2015/01/01}{\@xtryfc}%
874 <latexrelease>                                     {float order in 2-column}%
875 <*2ekernel|latexrelease>
876 \def\@xtryfc #1{%
877   \@next\reserved@a\@trylist{}{}%
878   \@currtype \count #1%
879   \divide\@currtype\@xxxii
880   \multiply\@currtype\@xxxii
881   \@bitor \@currtype \@failedlist
882   \@testfp #1%
883   \@testwrongwidth #1%
884   \ifdim \ht #1>\@colht
885     \@testtrue
886   \fi
887   \if@test
888     \@cons\@failedlist #1%
889   \else
890     \@ytryfc #1%
891   \fi}%
892 </2ekernel|latexrelease>
893 <latexrelease>\EndIncludeInRelease
894 <latexrelease>\IncludeInRelease{0000/00/00}{\@xtryfc}%
895 <latexrelease>                                     {float order in 2-column}%
896 <latexrelease>\def\@xtryfc #1{%
897 <latexrelease>   \@next\reserved@a\@trylist{}{}%
898 <latexrelease>   \@currtype \count #1%
899 <latexrelease>   \divide\@currtype\@xxxii
900 <latexrelease>   \multiply\@currtype\@xxxii
901 <latexrelease>   \@bitor \@currtype \@failedlist
902 <latexrelease>   \@testfp #1%
903 <latexrelease>   \ifdim \ht #1>\@colht
904 <latexrelease>     \@testtrue
905 <latexrelease>   \fi
906 <latexrelease>   \if@test
907 <latexrelease>     \@cons\@failedlist #1%
908 <latexrelease>   \else
909 <latexrelease>     \@ytryfc #1%
910 <latexrelease>   \fi}%
911 <latexrelease>\EndIncludeInRelease
912 <*2ekernel>
```

\@ytryfc

```
913 \def\@ytryfc #1{%
914   \begingroup
915   \gdef\@flsucceed{\@elt #1}%
916   \global\let\@flfail\@empty
917   \@tempdima\ht #1%
918   \let\@elt\@ztryfc
919   \@trylist
920   \ifdim \@tempdima >\@fpmin
921     \global\@fcolmadetrue
```

```

922 \else
923 \cons\@failedlist #1%
924 \fi
925 \endgroup
926 \if@fcolmade
927 \let\@elt\@gobble
928 \fi}

\@ztryfc
929 (/2ekernel)
930 \latexrelease\IncludeInRelease{2015/01/01}\@ztryfc}%
931 \latexrelease\float order in 2-column}%
932 (*2ekernel| latexrelease)
933 \def\@ztryfc #1{%
934 \tempcnta\count #1%
935 \divide\tempcnta\xxxii
936 \multiply\tempcnta\@xxxii
937 \bitor \tempcnta {\@failedlist \flfail}%
938 \testfp #1%
939 \testwrongwidth #1%
940 \tempdimb\tempdima
941 \advance\tempdimb\ht #1%
942 \advance\tempdimb\fpsep
943 \ifdim \tempdimb >\colht
944 \testtrue
945 \fi
946 \if@test
947 \cons\flfail #1%
948 \else
949 \cons\flsucceed #1%
950 \tempdima\tempdimb
951 \fi}%
952 (/2ekernel| latexrelease)
953 \latexrelease\EndIncludeInRelease
954 \latexrelease\IncludeInRelease{0000/00/00}\@ztryfc}%
955 \latexrelease\float order in 2-column}%
956 \latexrelease\def\@ztryfc #1{%
957 \latexrelease\tempcnta \count#1%
958 \latexrelease\divide\tempcnta\@xxxii
959 \latexrelease\multiply\tempcnta\@xxxii
960 \latexrelease\bitor \tempcnta {\@failedlist \flfail}%
961 \latexrelease\testfp #1%
962 \latexrelease\tempdimb\tempdima
963 \latexrelease\advance\tempdimb \ht#1%
964 \latexrelease\advance\tempdimb\fpsep
965 \latexrelease\ifdim \tempdimb >\colht
966 \latexrelease\testtrue
967 \latexrelease\fi
968 \latexrelease\if@test
969 \latexrelease\cons\flfail #1%
970 \latexrelease\else
971 \latexrelease\cons\flsucceed #1%

```



```

972 \latexrelease) \tempdima\tempdimb
973 \latexrelease) \fi}%
974 \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.

```

975 (*2kernel | fltrace)
976 \def \@addtobot {%
977 (*trace)
978   \fl@trace{***Start addtobot}%
979 /trace)
980   \@getfpsbit 4\relax
981 (*trace)
982   \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi bot:
983                                           \the \@fpstype}%
984 /trace)
985   \ifodd \@tempcnta
986     \@flsetnum \@botnum
987     \ifnum \@botnum>\z@
988       \@tempswafalse
989       \@flcheckspace \@botroom \@botlist
990       \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?

```

991     \global \maxdepth \z@
992     \@flupdates \@botnum \@botroom \@botlist
993 (*trace)
994     \fl@trace{colroom (after-bot) = \the \@colroom}%
995     \fl@trace{colnum (after-bot) = \the \@colnum}%
996     \fl@trace{botnum (after-bot) = \the \@botnum}%
997     \fl@trace{***Success: bot}%
998 /trace)
999     \inserttrue
1000   \fi
1001 (*trace)
1002   \else
1003     \fl@trace{Fail: botnum = \the \@botnum:
1004                                           fpstype \the \@fpstype=ORD?}%
1005     \ifnum \@fpstype<\sixt@n
1006       \fl@trace{ERROR: !b float not successful (addtobot)}%
1007     \fi
1008 /trace)
1009   \fi
1010   \fi
1011 }

```

`\@addtotoporbot` Lots of changes.

```

1012 \def \@addtotoporbot {%
1013 (*trace)
1014   \fl@trace{***Start addtotoporbot}%
1015 /trace)

```

```

1016 \getfpsbit \tw@
1017 (*trace)
1018 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi top:
1019 \the \@fpstype}%
1020 /trace)
1021 \ifodd \@tempcnta
1022 \flsetnum \@topnum
1023 \ifnum \@topnum>\z@
1024 \@tempwafalse
1025 \flcheckspace \@toproom \@toplist
1026 \if@tempswa
1027 \bitor\@currtype{\@midlist\@botlist}%
1028 (*trace)
1029 \fl@trace{(mid+bot)list: \@midlist, \@botlist:
1030 (addtotoporbot-before)}%
1031 /trace)
1032 \if@test
1033 (*trace)
1034 \fl@trace{type already on list: mid or bot---sent to addtobot}%
1035 /trace)
1036 \else
1037 \flupdates \@topnum \@toproom \@toplist
1038 (*trace)
1039 \fl@trace{colroom (after-top) = \the \@colroom}%
1040 \fl@trace{colnum (after-top) = \the \@colnum}%
1041 \fl@trace{topnum (after-top) = \the \@topnum}%
1042 \fl@trace{***Success: top}%
1043 /trace)
1044 \@inserttrue
1045 \fi
1046 \fi
1047 (*trace)
1048 \else
1049 \fl@trace{Fail: topnum = \the \@topnum: fpstype
1050 \the \@fpstype=ORD?}%
1051 \ifnum \@fpstype<\sixt@n
1052 \fl@trace{ERROR: !t float not successful (addtotoporbot)}%
1053 \fi
1054 /trace)
1055 \fi
1056 \fi
1057 \if@insert
1058 \else
1059 (*trace)
1060 \fl@trace{sent to addtobot (addtotoporbot)}%
1061 /trace)
1062 \@addtobot
1063 \fi
1064 }
1065 /2kernel | fltrace)

```

\@addtocurcol Lots of changes.

```

1066 (latexrelease | fltrace | flafter)\IncludeInRelease{2015/01/01}%
1067 (latexrelease | fltrace | flafter) {\@addtocurcol}{float order in 2-column}%

```

```

1068 <*2ekernel | latexrelease | fltrace | flafter>
1069 \def \@addtocurcol {%
1070 <*trace>
1071   \fl@trace{***Start addtocurcol}%
1072 </trace>
1073   \@insertfalse
1074   \@setfloattypecounts
1075   \ifnum \@fpstype=8
1076 <*trace>
1077     \fl@trace{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
1078 </trace>
1079   \else
1080     \ifnum \@fpstype=24
1081 <*trace>
1082       \fl@trace{fpstype p only (addtocurcol): \the \@fpstype = 24?}%
1083 </trace>
1084   \else
1085     \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.

```

1086 <*trace>
1087   \fl@trace{textfloatsheight (before) = \the \@textfloatsheight}%
1088 </trace>
1089   \advance \@textmin \@textfloatsheight
1090   \@reqcolroom \@pageht

```

This line must be removed since \@specialoutput changed.

```

1091 %       \advance \@reqcolroom \@pagedp
1092 <*trace>
1093   \fl@trace{textmin + textfloatsheight: \the \@textmin}%
1094   \fl@trace{page-so-far: \the \@reqcolroom}%
1095 </trace>
1096   \ifdim \@textmin>\@reqcolroom
1097     \@reqcolroom \@textmin
1098 <*trace>
1099   \fl@trace{ORD? textmin being used}%
1100 </trace>
1101   \fi
1102   \advance \@reqcolroom \ht\@currbox
1103 <*trace>
1104   \fl@trace{float size = \the \ht \@currbox (addtocurcol)}%
1105   \fl@trace{colroom = \the \@colroom (addtocurcol)}%
1106   \fl@trace{reqcolroom = \the \@reqcolroom (addtocurcol)}%
1107 </trace>
1108   \ifdim \@colroom>\@reqcolroom
1109     \flsetnum \@colnum
1110     \ifnum \@colnum>\z@
1111       \bitor\@currtype\@deferlist

```

We need to defer the float also if its width doesn't fit.

```

1112   \@testwrongwidth\@currbox

```

```

1113 (*trace)
1114 \fl@trace{deferlist: \@deferlist: (addtocurcol-before)}%
1115 (/trace)
1116 \if@test
1117 (*trace)
1118 \fl@trace{type already on list: defer (addtocurcol)}%
1119 (/trace)
1120 \else
1121 \@bitor\@currtype\@botlist
1122 (*trace)
1123 \fl@trace{botlist: \@botlist: (addtocurcol-before)}%
1124 (/trace)
1125 \if@test
1126 (*trace)
1127 \fl@trace{type already on list: bot---sent to addtobot}%
1128 (/trace)
1129 \@addtobot
1130 \else
1131 (*trace)
1132 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi
1133 here: \the \@fpstype}%
1134 (/trace)
1135 \ifodd \count\@currbox
1136 \advance \@reqcolroom \intextsep
1137 \ifdim \@colroom>\@reqcolroom
1138 \global \advance \@colnum \m@ne
1139 \global \advance \@textfloatsheight \ht\@currbox

```

This may sometimes give an overestimate.

```

1140 \global \advance \@textfloatsheight 2\intextsep
1141 \@cons \@midlist \@currbox
1142 (*trace)
1143 \fl@trace{***Success: here}%
1144 \fl@trace{textfloatsheight (after-here) =
1145 \the \@textfloatsheight}%
1146 \fl@trace{colnum (after-here) = \the \@colnum}%
1147 (/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).

```

1148 \if@nobreak
1149 \nobreak
1150 \@nobreakfalse

```

```

1151             \everypar{}%
1152         \else
1153             \addpenalty \interlinepenalty
1154         \fi
1155         \vskip \intextsep
1156         \box\@currbox
1157         \penalty\interlinepenalty
1158         \vskip\intextsep
1159         \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi

Typesetting ends here.

1160         \outputpenalty \z@
1161         \@inserttrue
1162 (*trace)
1163         \else
1164             \fl@trace{Fail---no room at 2nd test of colroom
1165                 (addtocorcol \string\intextsep)}%
1166         </trace>
1167         \fi
1168         \fi
1169         \if@insert
1170         \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 :- )

1171 (*2ekernel | fltrace | latexrelease)
1172 (*trace)
1173             \fl@trace{not here: sent to addtotoporbot}%
1174 </trace>
1175             \@addtotoporbot
1176 </2ekernel | fltrace | latexrelease>
1177 (*!2ekernel&!fltrace&!latexrelease)
1178 (*trace)
1179             \fl@trace{not here: sent to addtobot}%
1180 </trace>
1181             \@addtobot
1182 </!2ekernel&!fltrace&!latexrelease>
1183             \fi
1184             \fi
1185             \fi
1186 (*trace)
1187         \else
1188             \fl@trace{Fail: colnum = \the \@colnum:
1189                 fpstype \the \@fpstype=ORD?}%
1190             \ifnum \@fpstype<\sist@n
1191                 \fl@trace{ERROR: BANG float not successful (addtocurcol)}%
1192             \fi
1193 </trace>
1194             \fi
1195 (*trace)
1196         \else
1197             \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1198                 (addtocurcol)}%
1199 </trace>

```

```

1200     \fi
1201     \fi
1202     \fi
1203     \if@insert
1204     \else
1205         \@resetfps
1206     (*trace)
1207         \fl@trace{put on deferlist (addtocurcol)}%
1208     (/trace)
1209     \@cons\@deferlist\@currbox
1210 (*trace)
1211     \fl@trace{deferlist: \@deferlist: (addtocurcol-after)}%
1212 (/trace)
1213     \fi
1214 }%
1215 (/2kernel | latexrelease | fltrace | flafter)
1216 (latexrelease | fltrace | flafter)\EndIncludeInRelease
1217 (latexrelease | fltrace | flafter)\IncludeInRelease{0000/00/00}%
1218 (latexrelease | fltrace | flafter) {\@addtocurcol}{float order in 2-column}%
1219 (latexrelease | fltrace | flafter)\def \@addtocurcol {%
1220 (*trace)
1221 (latexrelease | fltrace | flafter) \fl@trace{***Start addtocurcol}%
1222 (/trace)
1223 (latexrelease | fltrace | flafter) \@insertfalse
1224 (latexrelease | fltrace | flafter) \@setfloattypescounts
1225 (latexrelease | fltrace | flafter) \ifnum \@fpstype=8
1226 (*trace)
1227 (latexrelease | fltrace | flafter) \fl@trace{fpstype !p only (addtocurcol):
1228 (latexrelease | fltrace | flafter) \the \@fpstype = 8?}%
1229 (/trace)
1230 (latexrelease | fltrace | flafter) \else
1231 (latexrelease | fltrace | flafter) \ifnum \@fpstype=24
1232 (*trace)
1233 (latexrelease | fltrace | flafter) \fl@trace{fpstype p only (addtocurcol):
1234 (latexrelease | fltrace | flafter) \the \@fpstype = 24?}%
1235 (/trace)
1236 (latexrelease | fltrace | flafter) \else
1237 (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.

```

1238 (*trace)
1239 (latexrelease | fltrace | flafter) \fl@trace{textfloatsheight (before) =
1240 (latexrelease | fltrace | flafter) \the \@textfloatsheight}%
1241 (/trace)
1242 (latexrelease | fltrace | flafter) \advance \@textmin \@textfloatsheight
1243 (latexrelease | fltrace | flafter) \@reqcolroom \@pageht

```

This line must be removed since `\@specialoutput` changed.

```

1244 % \advance \@reqcolroom \@pagedp
1245 (*trace)
1246 (latexrelease | fltrace | flafter) \fl@trace{textmin + textfloatsheight:
1247 (latexrelease | fltrace | flafter) \the \@textmin}%

```

1248	(latexrelease fltrace flafter)	\fl@trace{page-so-far: \the \@reqcolroom}%
1249	(latexrelease fltrace flafter)	
1250	/trace)	
1251	(latexrelease fltrace flafter)	\ifdim \@textmin>\@reqcolroom
1252	(latexrelease fltrace flafter)	\@reqcolroom \@textmin
1253	(*trace)	
1254	(latexrelease fltrace flafter)	\fl@trace{ORD? textmin being used}%
1255	/trace)	
1256	(latexrelease fltrace flafter)	\fi
1257	(latexrelease fltrace flafter)	\advance \@reqcolroom \ht\@currbox
1258	(*trace)	
1259	(latexrelease fltrace flafter)	\fl@trace{float size =
1260	(latexrelease fltrace flafter)	\the \ht \@currbox (addtocurcol))%
1261	(latexrelease fltrace flafter)	\fl@trace{colroom =
1262	(latexrelease fltrace flafter)	\the \@colroom (addtocurcol))%
1263	(latexrelease fltrace flafter)	\fl@trace{reqcolroom =
1264	(latexrelease fltrace flafter)	\the \@reqcolroom (addtocurcol))%
1265	/trace)	
1266	(latexrelease fltrace flafter)	\ifdim \@colroom>\@reqcolroom
1267	(latexrelease fltrace flafter)	\@flsetnum \@colnum
1268	(latexrelease fltrace flafter)	\ifnum \@colnum>\z@
1269	(latexrelease fltrace flafter)	\@bitor\@currtype\@deferlist
1270	(*trace)	
1271	(latexrelease fltrace flafter)	\fl@trace{deferlist:
1272	(latexrelease fltrace flafter)	\@deferlist: (addtocurcol-before))%
1273	/trace)	
1274	(latexrelease fltrace flafter)	\if@test
1275	(*trace)	
1276	(latexrelease fltrace flafter)	\fl@trace{type already on list:
1277	(latexrelease fltrace flafter)	defer (addtocurcol))%
1278	/trace)	
1279	(latexrelease fltrace flafter)	\else
1280	(latexrelease fltrace flafter)	\@bitor\@currtype\@botlist
1281	(*trace)	
1282	(latexrelease fltrace flafter)	\fl@trace{botlist: \@botlist:
1283	(latexrelease fltrace flafter)	(addtocurcol-before))%
1284	/trace)	
1285	(latexrelease fltrace flafter)	\if@test
1286	(*trace)	
1287	(latexrelease fltrace flafter)	\fl@trace{type already on list:
1288	(latexrelease fltrace flafter)	bot---sent to addtobot}%
1289	/trace)	
1290	(latexrelease fltrace flafter)	\@addtobot
1291	(latexrelease fltrace flafter)	\else
1292	(*trace)	
1293	(latexrelease fltrace flafter)	\fl@trace{fpstype
1294	(latexrelease fltrace flafter)	\ifodd \@tempcnta OK \else not \fi
1295	(latexrelease fltrace flafter)	here: \the \@fpstype}%
1296	/trace)	
1297	(latexrelease fltrace flafter)	\ifodd \count\@currbox
1298	(latexrelease fltrace flafter)	\advance \@reqcolroom \intextsep
1299	(latexrelease fltrace flafter)	\ifdim \@colroom>\@reqcolroom
1300	(latexrelease fltrace flafter)	\global \advance \@colnum \m@ne
1301	(latexrelease fltrace flafter)	\global \advance

```

1302 <latexrelease | fltrace | flafter>          \@textfloatsheight\ht\@currbox
This may sometimes give an overestimate.
1303 <latexrelease | fltrace | flafter>          \global \advance
1304 <latexrelease | fltrace | flafter>          \@textfloatsheight 2\intextsep
1305 <latexrelease | fltrace | flafter>          \@cons \@midlist \@currbox
1306 <*trace>
1307 <latexrelease | fltrace | flafter>          \fl@trace{***Success: here}%
1308 <latexrelease | fltrace | flafter>          \fl@trace{textfloatsheight
1309 <latexrelease | fltrace | flafter>              (after-here) =
1310 <latexrelease | fltrace | flafter>              \the \@textfloatsheight}%
1311 <latexrelease | fltrace | flafter>          \fl@trace{colnum (after-here) =
1312 <latexrelease | fltrace | flafter>              \the \@colnum}%
1313 </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).
1314 <latexrelease | fltrace | flafter>          \if@nobreak
1315 <latexrelease | fltrace | flafter>          \nobreak
1316 <latexrelease | fltrace | flafter>          \@nobreakfalse
1317 <latexrelease | fltrace | flafter>          \everypar{}%
1318 <latexrelease | fltrace | flafter>          \else
1319 <latexrelease | fltrace | flafter>          \addpenalty\interlinepenalty
1320 <latexrelease | fltrace | flafter>          \fi
1321 <latexrelease | fltrace | flafter>          \vskip \intextsep
1322 <latexrelease | fltrace | flafter>          \box\@currbox
1323 <latexrelease | fltrace | flafter>          \penalty\interlinepenalty
1324 <latexrelease | fltrace | flafter>          \vskip\intextsep
1325 <latexrelease | fltrace | flafter>          \ifnum\outputpenalty
1326 <latexrelease | fltrace | flafter>              <-\@Mii \vskip
1327 <latexrelease | fltrace | flafter>          -\parskip\fi
Typesetting ends here.
1328 <latexrelease | fltrace | flafter>          \outputpenalty \z@
1329 <latexrelease | fltrace | flafter>          \inserttrue
1330 <*trace>
1331 <latexrelease | fltrace | flafter>          \else
1332 <latexrelease | fltrace | flafter>          \fl@trace{Fail---no room at 2nd test of colroom
1333 <latexrelease | fltrace | flafter>              (addtocorcol \string\intextsep)}%
1334 </trace>
1335 <latexrelease | fltrace | flafter>          \fi
1336 <latexrelease | fltrace | flafter>          \fi
1337 <latexrelease | fltrace | flafter>          \if@insert

```



```

1338 <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 :-)
1339 <*2ekernel | fltrace>
1340 <*trace>
1341 <latexrelease | fltrace | flafter>          \fl@trace{not here: sent to addtotoporbot}%
1342 </trace>
1343 <latexrelease | fltrace | flafter>          \@addtotoporbot
1344 </2ekernel | fltrace>
1345 <*!2ekernel&!autoload&!fltrace>
1346 <*trace>
1347 <latexrelease | fltrace | flafter>          \fl@trace{not here: sent to addtobot}%
1348 </trace>
1349 <latexrelease | fltrace | flafter>          \@addtobot
1350 </!2ekernel&!autoload&!fltrace>
1351 <latexrelease | fltrace | flafter>          \fi
1352 <latexrelease | fltrace | flafter>          \fi
1353 <latexrelease | fltrace | flafter>          \fi
1354 <*trace>
1355 <latexrelease | fltrace | flafter>          \else
1356 <latexrelease | fltrace | flafter>          \fl@trace{Fail: colnum = \the \@colnum:
1357 <latexrelease | fltrace | flafter>          fpstype \the \@fpstype=ORD?}%
1358 <latexrelease | fltrace | flafter>          \ifnum \@fpstype<\sist@n
1359 <latexrelease | fltrace | flafter> \fl@trace{ERROR: BANG float not successful
1360 <latexrelease | fltrace | flafter>          (addtocurcol)}}%
1361 <latexrelease | fltrace | flafter>          \fi
1362 </trace>
1363 <latexrelease | fltrace | flafter>          \fi
1364 <*trace>
1365 <latexrelease | fltrace | flafter>          \else
1366 <latexrelease | fltrace | flafter>          \fl@trace{Fail---no room: fl box ht:
1367 <latexrelease | fltrace | flafter>          \the \ht \@currbox (addtocurcol)}}%
1368 </trace>
1369 <latexrelease | fltrace | flafter>          \fi
1370 <latexrelease | fltrace | flafter>          \fi
1371 <latexrelease | fltrace | flafter>          \fi
1372 <latexrelease | fltrace | flafter>          \if@insert
1373 <latexrelease | fltrace | flafter>          \else
1374 <latexrelease | fltrace | flafter>          \@resetfps
1375 <*trace>
1376 <latexrelease | fltrace | flafter>          \fl@trace{put on deferlist (addtocurcol)}}%
1377 </trace>
1378 <latexrelease | fltrace | flafter>          \@cons\@deferlist\@currbox
1379 <*trace>
1380 <latexrelease | fltrace | flafter>          \fl@trace{deferlist: \@deferlist:
1381 <latexrelease | fltrace | flafter>          (addtocurcol-after)}}%
1382 </trace>
1383 <latexrelease | fltrace | flafter>          \fi
1384 <latexrelease | fltrace | flafter>          }%
1385 <latexrelease | fltrace | flafter>\EndIncludeInRelease

```

\@addtonextcol Lots of changes.

```

1386 \latexrelease | fltrace \IncludeInRelease{2015/01/01}
1387 \latexrelease | fltrace \{@addtonextcol\}{float order in 2-column}%
1388 \*2ekernel | latexrelease | fltrace)
1389 \def\@addtonextcol{%
1390   \begingroup
1391   \*trace)
1392   \fl@trace{***Start addtonextcol}%
1393 \trace)
1394   \insertfalse
1395   \setfloattypecounts
1396   \ifnum \@fpstype=8
1397 \*trace)
1398   \fl@trace{fpstype not curcol: \the \@fpstype = 8?}%
1399 \trace)
1400   \else
1401   \ifnum \@fpstype=24
1402 \*trace)
1403   \fl@trace{fpstype not curcol: \the \@fpstype = 24?}%
1404 \trace)
1405   \else
1406   \flsettextmin
1407 \*trace)
1408   \fl@trace{text-so-far: Opt (top of col)}%
1409 \trace)
1410   \@reqcolroom \ht\@currbox
1411 \*trace)
1412   \fl@trace{float size: \the \@reqcolroom (addtonextcol)}%
1413 \trace)
1414   \advance \@reqcolroom \@textmin
1415 \*trace)
1416   \fl@trace{colroom = \the \@colroom (addtonextcol)}%
1417   \fl@trace{reqcolroom = \the \@reqcolroom (addtonextcol)}%
1418 \trace)
1419   \ifdim \@colroom>\@reqcolroom
1420   \flsetnum \@colnum
1421   \ifnum \@colnum>\z@
1422   \bitor\@currtype\@deferlist
1423 \*trace)
1424   \fl@trace{deferlist: \@deferlist: (addtonextcol-before)}%
1425 \trace)
1426   \@testwrongwidth\@currbox
1427   \if@test
1428 \*trace)
1429   \fl@trace{type already on list: defer (addtonextcol)}%
1430 \trace)
1431   \else
1432 \*trace)
1433   \fl@trace{sent to addtotoporbot (addtonextcol)}%
1434 \trace)
1435   \@addtotoporbot
1436   \fi
1437 \fi
1438 \*trace)

```

```

1439         \else
1440             \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1441                                     (addtonextcol)}%
1442         /trace>
1443         \fi
1444         \fi
1445         \fi
1446         \if@insert
1447         \else
1448         (*trace)
1449             \fl@trace{put back on deferlist (addtonextcol)}%
1450         /trace>
1451             \@cons\@deferlist\@currbox
1452         (*trace)
1453             \fl@trace{deferlist: \@deferlist: (addtonextcol-after)}%
1454         /trace>
1455         \fi
1456         (*trace)
1457             \fl@trace{End of addtonextcol -- locally counts:}%
1458             \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1459         /trace>
1460         \endgroup
1461         (*trace)
1462             \fl@trace{End of addtonextcol -- globally counts:}%
1463             \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1464         /trace>
1465     }%
1466 /2ekernel | latexrelease | fltrace>
1467 /latexrelease | fltrace>\EndIncludeInRelease
1468 /latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
1469 /latexrelease | fltrace> {\@addtonextcol}{float order in 2-column}%
1470 /latexrelease | fltrace>\def\@addtonextcol{%
1471 /latexrelease | fltrace> \begingroup
1472 (*trace)
1473 /latexrelease | fltrace> \fl@trace{***Start addtonextcol}%
1474 /trace>
1475 /latexrelease | fltrace> \@insertfalse
1476 /latexrelease | fltrace> \@setfloattypescounts
1477 /latexrelease | fltrace> \ifnum \@fpstype=8
1478 (*trace)
1479 /latexrelease | fltrace> \fl@trace{fpstype not curcol:
1480 /latexrelease | fltrace> \the \@fpstype = 8?}%
1481 /trace>
1482 /latexrelease | fltrace> \else
1483 /latexrelease | fltrace> \ifnum \@fpstype=24
1484 (*trace)
1485 /latexrelease | fltrace> \fl@trace{fpstype not curcol:
1486 /latexrelease | fltrace> \the \@fpstype = 24?}%
1487 /trace>
1488 /latexrelease | fltrace> \else
1489 /latexrelease | fltrace> \@flsettextmin
1490 (*trace)
1491 /latexrelease | fltrace> \fl@trace{text-so-far: Opt (top of col)}%
1492 /trace>

```

```

1493 \latexrelease | fltrace> \@reqcolroom \ht\@currbox
1494 \*trace>
1495 \latexrelease | fltrace> \fl@trace{float size:
1496 \latexrelease | fltrace> \the \@reqcolroom (addtonextcol)}%
1497 \latexrelease | fltrace>
1498 \trace>
1499 \latexrelease | fltrace> \advance \@reqcolroom \@textmin
1500 \*trace>
1501 \latexrelease | fltrace> \fl@trace{colroom =
1502 \latexrelease | fltrace> \the \@colroom (addtonextcol)}%
1503 \latexrelease | fltrace> \fl@trace{reqcolroom =
1504 \latexrelease | fltrace> \the \@reqcolroom (addtonextcol)}%
1505 \trace>
1506 \latexrelease | fltrace> \ifdim \@colroom>\@reqcolroom
1507 \latexrelease | fltrace> \@flsetnum \@colnum
1508 \latexrelease | fltrace> \ifnum\@colnum>\z@
1509 \latexrelease | fltrace> \@bitor\@currtype\@deferlist
1510 \*trace>
1511 \latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1512 \latexrelease | fltrace> (addtonextcol-before)}%
1513 \trace>
1514 \latexrelease | fltrace> \if@test
1515 \*trace>
1516 \latexrelease | fltrace> \fl@trace{type already on list:
1517 \latexrelease | fltrace> defer (addtonextcol)}%
1518 \trace>
1519 \latexrelease | fltrace> \else
1520 \*trace>
1521 \latexrelease | fltrace> \fl@trace{sent to addtotoporbot
1522 \latexrelease | fltrace> (addtonextcol)}%
1523 \trace>
1524 \latexrelease | fltrace> \@addtotoporbot
1525 \latexrelease | fltrace> \fi
1526 \latexrelease | fltrace> \fi
1527 \*trace>
1528 \latexrelease | fltrace> \else
1529 \latexrelease | fltrace> \fl@trace{Fail---no room: fl box ht:
1530 \latexrelease | fltrace> \the \ht \@currbox (addtonextcol)}%
1531 \trace>
1532 \latexrelease | fltrace> \fi
1533 \latexrelease | fltrace> \fi
1534 \latexrelease | fltrace> \fi
1535 \latexrelease | fltrace> \if@insert
1536 \latexrelease | fltrace> \else
1537 \*trace>
1538 \latexrelease | fltrace> \fl@trace{put back on deferlist
1539 \latexrelease | fltrace> (addtonextcol)}%
1540 \trace>
1541 \latexrelease | fltrace> \@cons\@deferlist\@currbox
1542 \*trace>
1543 \latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1544 \latexrelease | fltrace> (addtonextcol-after)}%
1545 \trace>
1546 \latexrelease | fltrace> \fi

```

```

1547 (*trace)
1548 (latexrelease|fltrace) \fl@trace{End of addtonextcol --
1549 (latexrelease|fltrace) locally counts:}%
1550 (latexrelease|fltrace) \fl@trace{col: \the \@colnum.
1551 (latexrelease|fltrace) top: \the \@topnum. bot: \the \@botnum.}%
1552 (/trace)
1553 (latexrelease|fltrace) \endgroup
1554 (*trace)
1555 (latexrelease|fltrace) \fl@trace{End of addtonextcol --
1556 (latexrelease|fltrace) globally counts:}%
1557 (latexrelease|fltrace) \fl@trace{col: \the \@colnum.
1558 (latexrelease|fltrace) top: \the \@topnum. bot: \the \@botnum.}%
1559 (/trace)
1560 (latexrelease|fltrace)}%
1561 (latexrelease|fltrace)\EndIncludeInRelease

```

\@addtodblcol Lots of changes.

```

1562 (latexrelease|fltrace)\IncludeInRelease{2015/01/01}%
1563 (latexrelease|fltrace) {\@addtodblcol}{float order in 2-column}%
1564 (*2ekernel|latexrelease|fltrace)
1565 \def\@addtodblcol{%
1566   \begingroup
1567   (*trace)
1568   \fl@trace{***Start addtodblcol}%
1569 (/trace)
1570   \@insertfalse
1571   \@setfloatypecounts
1572   \@getfpsbit \tw@
1573 (*trace)
1574   \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi dbltop:
1575   \the \@fpstype}%
1576 (/trace)
1577   \ifodd\@tempcnta
1578     \@flsetnum \@dbltopnum
1579     \ifnum \@dbltopnum>\z@
1580       \@tempswafalse
1581       \ifdim \@dbltoproom>\ht\@currbox
1582         \@tempswatrue
1583 (*trace)
1584         \fl@trace{Space OK: \@dbltoproom =
1585         \the \@dbltoproom > \the \ht \@currbox
1586         (dbltoproom)}%
1587 (/trace)
1588   \else
1589 (*trace)
1590     \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
1591 (/trace)
1592     \ifnum \@fpstype<\sist@n
1593 (*trace)
1594         \fl@trace{BANG float ignoring \@dbltoproom}%
1595         \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1596         Ht float: \the \ht \@currbox-BANG}%
1597 (/trace)

```

Need to check that there is room on the page, using the local value of \@textmin

to make the necessary adjustment to \@dbltoproom.

```

1598      \advance \@dbltoproom \@textmin
1599 (*trace)
1600      \fl@trace{Local value of texmin: \the\@textmin}%
1601      \fl@trace{\@spaces space on page = \the \@dbltoproom.
1602              Ht float: \the \ht \@currbox-BANG}%
1603 (/trace)
1604      \ifdim \@dbltoproom>\ht\@currbox
1605      \@tempwattrue
1606 (*trace)
1607      \fl@trace{Space OK BANG: space on page =
1608              \the \@dbltoproom > \the \ht \@currbox}%
1609      \else
1610      \fl@trace{fpstype: \the \@fpstype}%
1611      \fl@trace{Fail---no room dbltoproom-BANG?:}%
1612      \fl@trace{\@spaces space on page = \the \@dbltoproom.
1613              Ht float: \the \ht \@currbox}%
1614 (/trace)
1615      \fi
1616      \advance \@dbltoproom -\@textmin
1617 (*trace)
1618      \else
1619      \fl@trace{fpstype: \the \@fpstype}%
1620      \fl@trace{Fail---no room dbltoproom-ORD?:}%
1621      \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1622              Ht float: \the \ht \@currbox}%
1623 (/trace)
1624      \fi
1625      \fi
1626      \if@tempwa
1627      \@bitor \@currtype \@deferlist
1628 (*trace)
1629      \fl@trace{(dbl)deferlist: \@deferlist: (before)}%
1630 (/trace)
1631      not in fixfloats?
1632      \if@test
1633 (*trace)
1634      \fl@trace{type already on list: (dbl)defer}%
1635 (/trace)
1636      \else
1637      \@tempdima -\ht\@currbox
1638      \advance\@tempdima
1639      -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
1640              \dblfloatsep \fi
1641      \global \advance \@dbltoproom \@tempdima
1642      \global \advance \@colht \@tempdima
1643      \global \advance \@dbltopnum \m@ne
1644      \@cons \@dbltoplist \@currbox
1645 (*trace)
1646      \fl@trace{dbltopnum (after) = \the \@dbltopnum}%
1647      \fl@trace{***Success: dbltop}%
1648 (/trace)

```

```

1649             \@inserttrue
1650             \fi
1651         \fi
1652 (*trace)
1653     \else
1654         \fl@trace{Fail: dbltopnum = \the \@dbltopnum: fpstype
1655                                     \the \@fpstype=ORD?}%
1656         \ifnum \@fpstype<\sixt@@n
1657             \fl@trace{ERROR: !t float not successful (addtodblcol)}%
1658         \fi
1659 (/trace)
1660     \fi
1661 \fi
1662 \if@insert
1663 \else
1664 (*trace)
1665     \fl@trace{put on deferlist}%
1666 (/trace)
1667     \@cons\@deferlist\@currbox
1668 (*trace)
1669     \fl@trace{(dbl)deferlist: \@deferlist: (after)}%
1670 (/trace)
1671 \fi
1672 (*trace)
1673     \fl@trace{End of addtodblcol -- locally count:}%
1674     \fl@trace{dbltop: \the \@dbltopnum.}%
1675 (/trace)
1676 \endgroup
1677 (*trace)
1678     \fl@trace{End of addtodblcol -- globally count:}%
1679     \fl@trace{dbltop: \the \@dbltopnum.}%
1680 (/trace)
1681 }%
1682 (/2ekernel | latexrelease | fltrace)
1683 (latexrelease | fltrace)\EndIncludeInRelease
1684 (latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
1685 (latexrelease | fltrace) {\@addtodblcol}{float order in 2-column}%
1686 (latexrelease | fltrace)\def\@addtodblcol{%
1687 (latexrelease | fltrace) \begingroup
1688 (*trace)
1689 (latexrelease | fltrace) \fl@trace{***Start addtodblcol}%
1690 (/trace)
1691 (latexrelease | fltrace) \@insertfalse
1692 (latexrelease | fltrace) \@setfloattypecounts
1693 (latexrelease | fltrace) \@getfpsbit \tw@
1694 (*trace)
1695 (latexrelease | fltrace) \fl@trace{fpstype \ifodd \@tempcnta OK
1696 (latexrelease | fltrace) \else not \fi dbltop: \the \@fpstype}%
1697 (/trace)
1698 (latexrelease | fltrace) \ifodd\@tempcnta
1699 (latexrelease | fltrace) \@flsetnum \@dbltopnum
1700 (latexrelease | fltrace) \ifnum \@dbltopnum>\z@
1701 (latexrelease | fltrace) \@tempswafalse
1702 (latexrelease | fltrace) \ifdim \@dbltoproom>\ht\@currbox

```

```

1703 <latexrelease | fltrace>          \@tempswatrue
1704 <*trace>
1705 <latexrelease | fltrace>          \fl@trace{Space OK: \@dbltoproom =
1706 <latexrelease | fltrace>          \the \@dbltoproom > \the \ht \@currbox
1707 <latexrelease | fltrace>          (dbltoproom)}%
1708 </trace>
1709 <latexrelease | fltrace>          \else
1710 <*trace>
1711 <latexrelease | fltrace>          \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
1712 </trace>
1713 <latexrelease | fltrace>          \ifnum \@fpstype<\sixt@@n
1714 <*trace>
1715 <latexrelease | fltrace>          \fl@trace{BANG float ignoring \@dbltoproom}%
1716 <latexrelease | fltrace>          \fl@trace{\@spaces \@dbltoproom =
1717 <latexrelease | fltrace>          \the \@dbltoproom.
1718 <latexrelease | fltrace>          Ht float: \the \ht \@currbox-BANG}%
1719 </trace>

```

Need to check that there is room on the page, using the local value of \@textmin to make the necessary adjustment to \@dbltoproom.

```

1720 <latexrelease | fltrace>          \advance \@dbltoproom \@textmin
1721 <*trace>
1722 <latexrelease | fltrace>          \fl@trace{Local value of texmin: \the \@textmin}%
1723 <latexrelease | fltrace>          \fl@trace{\@spaces space on page =
1724 <latexrelease | fltrace>          \the \@dbltoproom.
1725 <latexrelease | fltrace>          Ht float: \the \ht \@currbox-BANG}%
1726 </trace>
1727 <latexrelease | fltrace>          \ifdim \@dbltoproom>\ht \@currbox
1728 <latexrelease | fltrace>          \@tempswatrue
1729 <*trace>
1730 <latexrelease | fltrace>          \fl@trace{Space OK BANG: space on page =
1731 <latexrelease | fltrace>          \the \@dbltoproom > \the \ht \@currbox}%
1732 <latexrelease | fltrace>          \else
1733 <latexrelease | fltrace>          \fl@trace{fpstype: \the \@fpstype}%
1734 <latexrelease | fltrace>          \fl@trace{Fail---no room dbltoproom-BANG?:}%
1735 <latexrelease | fltrace>          \fl@trace{\@spaces space on page =
1736 <latexrelease | fltrace>          \the \@dbltoproom.
1737 <latexrelease | fltrace>          Ht float: \the \ht \@currbox}%
1738 </trace>
1739 <latexrelease | fltrace>          \fi
1740 <latexrelease | fltrace>          \advance \@dbltoproom -\@textmin
1741 <*trace>
1742 <latexrelease | fltrace>          \else
1743 <latexrelease | fltrace>          \fl@trace{fpstype: \the \@fpstype}%
1744 <latexrelease | fltrace>          \fl@trace{Fail---no room dbltoproom-ORD?:}%
1745 <latexrelease | fltrace>          \fl@trace{\@spaces \@dbltoproom =
1746 <latexrelease | fltrace>          \the \@dbltoproom.
1747 <latexrelease | fltrace>          Ht float: \the \ht \@currbox}%
1748 </trace>
1749 <latexrelease | fltrace>          \fi
1750 <latexrelease | fltrace>          \fi
1751 <latexrelease | fltrace>          \if@tempswa
1752 <latexrelease | fltrace>          \@bitor \@currtypes \@dbldeferlist
1753 <*trace>

```



```

1754 \latexrelease|fltrace> \fl@trace{dbldeferlist:
1755 \latexrelease|fltrace> \@dbldeferlist: (before)}%
1756 \trace>
1757 \latexrelease|fltrace> \if@test
1758 \*trace>
1759 \latexrelease|fltrace> \fl@trace{type already on list: dbldefer}%
1760 \trace>
1761 \latexrelease|fltrace> \else
1762 \latexrelease|fltrace> \@tempdima -\ht\@currbox
1763 \latexrelease|fltrace> \advance\@tempdima
1764 \latexrelease|fltrace> -\ifx \@dbltoplist\@empty
1765 \latexrelease|fltrace> \dbltextfloatsep
1766 \latexrelease|fltrace> \else \dblfloatsep \fi
1767 \latexrelease|fltrace> \global \advance \@dbltoproom \@tempdima
1768 \latexrelease|fltrace> \global \advance \@colht \@tempdima
1769 \latexrelease|fltrace> \global \advance \@dbltopnum \m@ne
1770 \latexrelease|fltrace> \@cons \@dbltoplist \@currbox
1771 \*trace>
1772 \latexrelease|fltrace> \fl@trace{dbltopnum (after) =
1773 \latexrelease|fltrace> \the \@dbltopnum}%
1774 \latexrelease|fltrace> \fl@trace{***Success: dbltop}%
1775 \trace>
1776 \latexrelease|fltrace> \@inserttrue
1777 \latexrelease|fltrace> \fi
1778 \latexrelease|fltrace> \fi
1779 \*trace>
1780 \latexrelease|fltrace> \else
1781 \latexrelease|fltrace> \fl@trace{Fail: dbltopnum = \the \@dbltopnum:
1782 \latexrelease|fltrace> fpstype \the \@fpstype=ORD?}%
1783 \latexrelease|fltrace> \ifnum \@fpstype<\sixt@n
1784 \latexrelease|fltrace> \fl@trace{ERROR: !t float not successful
1785 \latexrelease|fltrace> (addtodblcol)}%
1786 \latexrelease|fltrace> \fi
1787 \trace>
1788 \latexrelease|fltrace> \fi
1789 \latexrelease|fltrace> \fi
1790 \latexrelease|fltrace> \if@insert
1791 \latexrelease|fltrace> \else
1792 \*trace>
1793 \latexrelease|fltrace> \fl@trace{put on dbldeferlist}%
1794 \trace>
1795 \latexrelease|fltrace> \@cons\@dbldeferlist\@currbox
1796 \*trace>
1797 \latexrelease|fltrace> \fl@trace{dbldeferlist: \@dbldeferlist: (after)}%
1798 \trace>
1799 \latexrelease|fltrace> \fi
1800 \*trace>
1801 \latexrelease|fltrace> \fl@trace{End of addtodblcol -- locally count:}%
1802 \latexrelease|fltrace> \fl@trace{ dbltop: \the \@dbltopnum.}%
1803 \trace>
1804 \latexrelease|fltrace> \endgroup
1805 \*trace>
1806 \latexrelease|fltrace> \fl@trace{End of addtodblcol -- globally count:}%
1807 \latexrelease|fltrace> \fl@trace{dbltop: \the \@dbltopnum.}%

```

```

1808 </trace>
1809 <latexrelease | fltrace>}%
1810 <latexrelease | fltrace>\EndIncludeInRelease

```

\@addmarginpar

```

1811 (*2ekernel)
1812 \def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
1813   \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
1814   \if@twocolumn
1815     \if@firstcolumn \@tempcnta\m@ne \fi
1816   \else
1817     \if@mparswitch
1818       \ifodd\c@page \else\@tempcnta\m@ne \fi
1819     \fi
1820     \if@reversemargin \@tempcnta -\@tempcnta \fi
1821   \fi
1822   \ifnum\@tempcnta <\z@ \global\setbox\@marbox\box\@currbox \fi
1823   \@tempdima\@mparbottom
1824   \advance\@tempdima -\@pageht
1825   \advance\@tempdima\ht\@marbox
1826   \ifdim\@tempdima >\z@
1827     \@latex@warning@no@line {Marginpar on page \thepage\space moved}%
1828   \else
1829     \@tempdima\z@
1830   \fi
1831   \global\@mparbottom\@pageht
1832   \global\advance\@mparbottom\@tempdima
1833   \global\advance\@mparbottom\dp\@marbox
1834   \global\advance\@mparbottom\marginparpush
1835   \advance\@tempdima -\ht\@marbox

```

Putting box movement inside the ‘marbox’:

```

1836 \global\setbox \@marbox
1837         \vbox {\vskip \@tempdima
1838             \box \@marbox}%
1839 \global \ht\@marbox \z@
1840 \global \dp\@marbox \z@

```

Sticking (rather than gluing:-) the ‘marbox’ to the line above, changed vskip to kern:

```

1841 \kern -\@pagedp
1842 \nointerlineskip
1843 \hb@xt@\columnwidth
1844   {\ifnum \@tempcnta >\z@
1845     \hskip\columnwidth \hskip\marginparsep
1846   \else
1847     \hskip -\marginparsep \hskip -\marginparwidth
1848   \fi
1849   \box\@marbox \hss}%

```

For this reason the following code can vanish:

```

\ nobreak           %% No longer needed.  CAR92/12
\ vskip -\@tempdima %% No longer needed.  CAR92/12

1850 \nointerlineskip
1851 \hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}

```

76.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for L^AT_EX2e.

`\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_EX do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```
1852 \newinsert \@kludgeins
1853 \global\dimen\@kludgeins \maxdimen
1854 \global\count\@kludgeins 1000
```

`\enlargethispage` The user command.

```
\enlargethispage* 1855 \gdef \enlargethispage {%
1856     \ifstar
1857     {%
1858     (*trace)
1859         \fl@trace{Enlarging page height * }%
1860     </trace>
1861         \@enlargepage{\hbox{\kern\p@}}}%
1862     {%
1863     (*trace)
1864         \fl@trace{Enlarging page height exactly---}%
1865     </trace>
1866         \@enlargepage\@empty}%
1867 }
```

`\@enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```
1868 \gdef\@enlargepage#1#2{%
1869 (*trace)
1870     \fl@trace{\@spaces\@spaces by #2}%
1871 </trace>
1872     \@tempkipa#2\relax
1873     \ifdim \@tempkipa>.5\maxdimen
```

```

1874     \@latexerr{Suggested\space extra\space height\space
1875               (\the\@tempskipa)\space dangerously\space
1876               large}\@eha
1877   \else
1878     \ifdim \vsize<.5\maxdimen
1879 (*trace)
1880     \fl@trace {Kludgeins added--pagegoal before: \the\pagegoal}%
1881 (/trace)
1882     \@bsphack
1883     \insert\@kludgeins{#1\vskip-\@tempskipa}%
1884     \@esphack

```

This next bit is for tracing only:

```

1885 (*trace)
1886     \ifvmode \par
1887     \fl@trace {Kludgeins added--pagegoal after: \the\pagegoal}%
1888     \fi
1889 (/trace)
1890   \else
1891     \@latexerr{Page\space height\space already\space
1892               too\space large}\@eha
1893   \fi
1894 \fi
1895 }
1896 (/2ekernel)

```

76.1.2 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^AT_EX2_ε.

```
\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 1897 \def \fl@trace
\fl@traceval 1898 \def \fl@tracemessage #1{\let\@elt\@empty\typeout{LaTeX2e: #1}}
\tracefloatvals 1899 \def \tracefloats{\let \fl@trace \fl@tracemessage}
\fl@tracemessage 1900 \def \tracefloatsoff {\let \fl@trace \@gobble}
1901 \tracefloatsoff
1902 \def \fl@traceval #1{\fl@trace{\string #1 = \the #1}}
1903 \IncludeInRelease{2015/01/01}{\tracefloatvals}%
1904 {trace float vals}%
1905 \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.¹³

```

1906 \begingroup
1907 \@dblfloatplacement
1908 \@floatplacement
1909 \fl@trace{***Float placement parameters:}%
1910 \fl@traceval\@colnum
1911 \fl@traceval\@colroom
1912 \fl@traceval\@topnum
1913 \fl@traceval\@toproom
1914 \fl@traceval\@botnum
1915 \fl@traceval\@botroom
1916 \fl@traceval\@fpmin
1917 \fl@trace{\string\textfraction = \textfraction}%
1918 \fl@traceval\@dbltopnum
1919 \fl@traceval\@dbltoproom
1920 \fl@trace{\string\textfraction = \textfraction}%
1921 \fl@trace{toplist: \@toplist}%
1922 \fl@trace{botlist: \@botlist}%
1923 \fl@trace{midlist: \@midlist}%
1924 \fl@trace{deferlist: \@deferlist}%
1925 \fl@trace{dbltoplist: \@dbltoplist}%
1926 %Fmi \fl@trace{dbldeferlist: \@dbldeferlist}%
1927 \endgroup
1928 }
1929 \EndIncludeInRelease
1930 \IncludeInRelease{0000/00/00}{\tracefloatvals}%
1931 {trace float vals}%
1932 \def \tracefloatvals{%
1933 \begingroup
1934 \@dblfloatplacement
1935 \@floatplacement
1936 \fl@trace{***Float placement parameters:}%
1937 \fl@traceval\@colnum
1938 \fl@traceval\@colroom
1939 \fl@traceval\@topnum

```

¹³This is a somewhat questionable design.

```

1940 \fl@traceval\@toproom
1941 \fl@traceval\@botnum
1942 \fl@traceval\@botroom
1943 \fl@traceval\@fpmin
1944 \fl@trace{\string\textfraction = \textfraction}%
1945 \fl@traceval\@dbltopnum
1946 \fl@traceval\@dbltoproom
1947 \fl@trace{\string\textfraction = \textfraction}%
1948 \fl@trace{toplist: \@toplist}%
1949 \fl@trace{botlist: \@botlist}%
1950 \fl@trace{midlist: \@midlist}%
1951 \fl@trace{deferlist: \@deferlist}%
1952 \fl@trace{dbltoplist: \@dbltoplist}%
1953 % next line only in old releases
1954 \fl@trace{dbldeferlist: \@dbldeferlist}%
1955 \endgroup
1956 }
1957 \EndIncludeInRelease

```

We need to make sure that `fltrace` comes before `flafter` to make the tracing work.

```

1958 \@ifpackageloaded{flafter}
1959 {
1960   \PackageWarningNoLine
1961     {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
              Attempting to recover by reloading 'flafter'}%

```

Hide the fact that `flafter` was already loaded and then request it anew.

```

1962 \expandafter\let\csname ver@flafter.sty\endcsname\relax
1963 \def\reserved@a#1{%
1964   \expandafter\let\csname\string#1+flafter+IIR\endcsname\relax}%
1965 \reserved@a\@addtocurcol
1966 \reserved@a\@addtonextcol
1967 \RequirePackage{flafter}}{}
1968 \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.

```

1969 (*flafter)
1970 \providecommand\fl@trace[1]{}
1971 \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.

```

1972 (*2kernel)
1973 \def \suppressfloats {%
1974   \ifnextchar [%
1975     \@flstop
1976     {\global \@colnum \z@}%
1977 }

```

Maybe this should be a loop over `#1`?

```

1978 \def \@flstop [#1]{%
1979   \if t#1%
1980     \global \@topnum \z@

```

```

1981 \fi
1982 \if b#1%
1983 \global \@botnum \z@
1984 \fi
1985 }

```

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`
`\@textfloatsheight` conservatism: it is possible that this could use a temporary register.

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.

```

1986 \newcount \@fpstype
1987 \newdimen \@reqcolroom
1988 \newdimen \@textfloatsheight
1989 \if2kernel

```

`\@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@\@captype \endcsname}%
  \edef \reserved@a {\the\temptokena}%
  \@onelevel@sanitize \reserved@a
  \edef \@fps {\@fps\reserved@a}%
}

```

```

1990 \if2kernel\iftraced
1991 \def \@fpsadddefault {%
1992 \iftraced
1993 \if@trace{fps changed from: \@fps}%
1994 \iftraced
1995 \edef \@fps {\@fps\csname fps@\@captype \endcsname}%
1996 \if@latex@warning {%
1997 No positions in optional float specifier.\MessageBreak
1998 Default added (so using '\@fps')}%
1999 }

```

`\@setfloattypecounts` Sets counters `\@fpstype` and `\@currtype`.
BANG == bit4 of `\count\@currbox` = 0.

```

2000 \def \@setfloattypecounts {%
2001 \if@currtype \count\@currbox
2002 \if@fpstype \count\@currbox
2003 \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
2004 \advance \@fpstype -\@currtype
2005 \iftraced
2006 \if@trace{(mod 32) fpstype: \the \@fpstype}%
2007 \if@trace{(mult of 32) currtype: \the \@currtype}%

```

```

2008 % Tracing only: but some should be changed into real errors/warnings?
2009 \ifnum \@fpstype<\sist@n
2010 \ifnum \@fpstype=\z@
2011 \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
2012 \fi
2013 \ifnum \@fpstype=\@ne
2014 \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 1?}%
2015 \fi
2016 \fl@trace{BANG float}%
2017 \else
2018 \ifnum \@fpstype=\sist@n
2019 \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
2020 \fi
2021 \ifnum \@fpstype=17
2022 \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 17?}%
2023 \fi
2024 \fl@trace{ORD float}%
2025 \fi
2026 \trace
2027 }
2028 \end{kernel} \end{fltrace}

```

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```

2029 (*2kernel)
2030 \def \@getfpsbit {%
2031 \boxfpsbit \@currbox
2032 }

```

`\@boxfpsbit` Used above.

```

2033 \def \@boxfpsbit #1#2{%
2034 \@tempcnta \count#1
2035 \divide \@tempcnta #2\relax
2036 }

```

`\@testfp` New definition of the float page test.

```

2037 \def \@testfp #1{%
2038 \@boxfpsbit #18\relax % Really ‘#1 8’ for human readers!
2039 \ifodd \@tempcnta
2040 \else
2041 \@testtrue
2042 \fi
2043 }

```

`\@setfpsbit` Sets required bit of `\@tempcnta` (to 1).

```

2044 \def \@setfpsbit #1{%
2045 \@tempcntb \@tempcnta
2046 \divide \@tempcntb #1\relax
2047 \ifodd \@tempcntb
2048 \else
2049 \advance \@tempcnta #1\relax
2050 \fi

```



```

2051 }
2052 </2ekernel>

```

\@resethfps 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.

```

2053 (*2ekernel | fltrace)
2054 \def \@resethfps {%
2055   \let\reserved@a\@empty
2056   \ifnum \@fpstype=\@ne
2057     \def \reserved@a {!}%
2058     \@fpstype 17
2059   \fi
2060   \ifnum \@fpstype=17
2061     \global \advance \count\@currbox \tw@
2062     \@latex@warning@no@line {%
2063       '\reserved@a h' float specifier changed to '\reserved@a ht'}%
2064 (*trace)
2065   \fl@trace{%
2066     't' added to '\reserved@a h'- new Count: \the \count\@currbox}%
2067 </trace>
2068   \fi
2069 }

```

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.

```

2070 \def \@flsetnum #1{%
2071 (*trace)
2072   \fl@trace{fpstype: \the \@fpstype (flsetnum \string#1)}%
2073 </trace>
2074   \ifnum \@fpstype<\sxt@
2075     \ifnum #1=\z@
2076 (*trace)
2077     \fl@trace{BANG float resetting \string#1 to 1}%
2078 </trace>
2079     #1\@ne
2080   \fi
2081 \fi
2082 (*trace)
2083   \fl@trace{#1 (before) = \the #1}%
2084 </trace>

```

2085 }

\@flsettextmin This ignores \textfraction space restriction in case BANG.

```

2086 \def \@flsettextmin {%
2087 (*trace)
2088   \fl@trace{fpstype: \the \@fpstype (flsettextmin)}%
2089 (/trace)
2090   \ifnum \@fpstype<\sixt@@n
2091 (*trace)
2092     \fl@trace{BANG ignoring textmin}%
2093 (/trace)
2094     \@textmin \z@
2095   \else
2096     \@textmin \textfraction\@colht
2097 (*trace)
2098     \fl@trace{ORD textmin = \the \@textmin}%
2099 (/trace)
2100   \fi
2101 }
```

\@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.

```

2102 \def \@flcheckspace #1#2{%
2103   \advance \@reqcolroom
2104   \ifx #2\@empty \textfloatsep \else \floatsep \fi
2105 (*trace)
2106   \fl@trace{colroom = \the \@colroom
2107                                     (flcheckspace \string#1 \string#2)}%
2108   \fl@trace{reqcolroom = \the \@reqcolroom
2109                                     (flcheckspace \string#1 \string#2)}%
2110 (/trace)
2111   \ifdim \@colroom>\@reqcolroom
2112     \ifdim #1>\ht\@currbox
2113       \@tempwattrue
2114 (*trace)
2115       \fl@trace{Space OK: #1 = \the #1 > \the \ht \@currbox
2116                                     (flcheckspace \string#1 \string#2)}%
2117 (/trace)
2118   \else
2119 (*trace)
2120     \fl@trace{fpstype: \the \@fpstype
2121                                     (flcheckspace \string#1 \string#2)}%
2122 (/trace)
2123     \ifnum \@fpstype<\sixt@@n
2124 (*trace)
2125       \fl@trace{BANG float ignoring #1
2126                                     (flcheckspace \string#1 \string#2):}%
2127       \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2128                                     BANG}%
2129 (/trace)
2130     \@tempwattrue
2131 (*trace)
```

```

2132     \else
2133     \fl@trace{Fail---no room (flcheckspace \string#1 \string#2)
2134             (fpstype \the \@fpstype=ORD?):}%
2135     \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2136             ORD?:}%
2137 \end{trace}
2138 \fi
2139 \fi
2140 \end{trace}
2141 \else
2142     \fl@trace{Fail---no room at 2nd test of colroom
2143             (flcheckspace \string#1 \string#2):}%
2144 \end{trace}
2145 \fi
2146 }
2147 \end{2kernel} \end{fltrace}

```

`\@flupdates` This updates everything when a float is placed.

```

2148 \end{2kernel}
2149 \def \@flupdates #1#2#3{%
2150   \global \advance #1\m@ne
2151   \global \advance \@colnum \m@ne
2152   \tempdima -\ht\@currbox
2153   \advance \@tempdima
2154   -\ifx #3\@empty \textfloatsep \else \floatsep \fi
2155   \global \advance #2\@tempdima
2156   \global \advance \@colroom \@tempdima
2157   \@cons #3\@currbox
2158 }
2159 \end{2kernel}

```

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 `\@docclearpage`?
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.

```
2160 <latexrelease | fltrace>\IncludeInRelease{2017/01/01}%
2161 <latexrelease | fltrace> {\@makefcolumn}{negative height floats}%
2162 <*2ekernel | fltrace | latexrelease>
2163 \def\@makefcolumn #1{%
2164   \begingroup
```


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.

```
2207 \setbox\@outputbox\vbox{\unvbox\@outputbox\unskip}%
2208 \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`.

```
2209 \toks@\expandafter{\topmark}%
2210 \xdef\@firstcoltopmark{\the\toks@}%
2211 \toks@\expandafter{\splitfirstmark}%
2212 \xdef\@firstcolfirstmark{\the\toks@}%
```

This test does not work if truly empty marks have been inserted, but L^AT_EX 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.)

```
2213 \ifx\@firstcolfirstmark\@empty
2214 \global\let\@setmarks\relax
2215 \else
2216 \gdef\@setmarks{%
2217 \let\firstmark\@firstcolfirstmark
2218 \let\topmark\@firstcoltopmark}%
2219 \fi
```

End of change

```
2220 \else
2221 \global\@firstcolumntrue
2222 \setbox\@outputbox\vbox{%
2223 \hb@xt@\textwidth{%
2224 \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
2225 \hfil
```

The color of the `\vrule` should be `\normalcolor` as to not inherit the color from the column.

```
2226 {\normalcolor\vrule \@width\columnseprule}%
2227 \hfil
2228 \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
2229 \fl@trace{PAGE: second column also boxed}%
2230 \@combinedblfloats
```

Override current first and top with those of first column if necessary

```
2231 \@setmarks
```

End of change

```
2232 \@outputpage
2233 \fl@trace{PAGE: two column page completed}%
2234 \begingroup
2235 \@dblfloatplacement
2236 \startdblcolumn
2237 \@whilesw\if@colmade \fi{\@outputpage
2238 \fl@trace{PAGE: double float page completed}%
2239 \startdblcolumn}%
2240 \endgroup
2241 \fi}%
```

```

2242 <latexrelease | fltrace>\EndIncludeInRelease
2243 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
2244 <latexrelease | fltrace> {\@outputdblcol}{2 column marks}%
2245 <latexrelease | fltrace>\def\@outputdblcol{%
2246 <latexrelease | fltrace> \if@firstcolumn
2247 <latexrelease | fltrace> \global \@firstcolumnfalse
2248 <latexrelease | fltrace> \global \setbox\@leftcolumn \box\@outputbox
2249 <*trace>
2250 <latexrelease | fltrace> \fl@trace{PAGE: first column boxed}%
2251 </trace>
2252 <latexrelease | fltrace> \else
2253 <latexrelease | fltrace> \global \@firstcolumntrue
2254 <latexrelease | fltrace> \setbox\@outputbox \vbox {%
2255 <latexrelease | fltrace> \hb@xt@\textwidth {%
2256 <latexrelease | fltrace> \hb@xt@\columnwidth {%
2257 <latexrelease | fltrace> \box\@leftcolumn \hss}%
2258 <latexrelease | fltrace> \hfil
2259 <latexrelease | fltrace> {\normalcolor\vrule
2260 <latexrelease | fltrace> \@width\columnseprule}%
2261 <latexrelease | fltrace> \hfil
2262 <latexrelease | fltrace> \hb@xt@\columnwidth {%
2263 <latexrelease | fltrace> \box\@outputbox \hss}%
2264 <latexrelease | fltrace> }%
2265 <latexrelease | fltrace> }%
2266 <*trace>
2267 <latexrelease | fltrace> \fl@trace{PAGE: second column also boxed}%
2268 </trace>
2269 <latexrelease | fltrace> \@combinedblfloats
2270 <latexrelease | fltrace> \@outputpage
2271 <*trace>
2272 <latexrelease | fltrace> \fl@trace{PAGE: two column page completed}%
2273 </trace>
2274 <latexrelease | fltrace> \begingroup
2275 <latexrelease | fltrace> \@dblfloatplacement
2276 <latexrelease | fltrace> \@startdblcolumn

This loop could be replaced by an \expandafter tail recursion in
\@startdblcolumn.

2277 <latexrelease | fltrace> \@whiles\if@fcolmade \fi
2278 <latexrelease | fltrace> {\@outputpage
2279 <*trace>
2280 <latexrelease | fltrace> \fl@trace{PAGE: double float page completed}%
2281 </trace>
2282 <latexrelease | fltrace> \@startdblcolumn}%
2283 <latexrelease | fltrace> \endgroup
2284 <latexrelease | fltrace> \fi
2285 <latexrelease | fltrace> }%
2286 <latexrelease | fltrace>\EndIncludeInRelease
2287 </2ekernel | fltrace | latexrelease>

```

76.1.3 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. 2288 (*2ekernel) 2289 <code>\newcount\c@topnumber</code> 2290 <code>\setcounter{topnumber}{2}</code>
<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. 2291 <code>\newcommand\topfraction{.7}</code>
<code>\c@bottomnumber</code>	This counter holds the maximum number of floats that can appear at the bottom of a text page or column. 2292 <code>\newcount\c@bottomnumber</code> 2293 <code>\setcounter{bottomnumber}{1}</code>
<code>\bottomfraction</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 bottom. 2294 <code>\newcommand\bottomfraction{.3}</code>
<code>\c@totalnumber</code>	This counter holds the maximum number of floats that can appear on any text page or column. 2295 <code>\newcount\c@totalnumber</code> 2296 <code>\setcounter{totalnumber}{3}</code>
<code>\textfraction</code>	This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text. 2297 <code>\newcommand\textfraction{.2}</code>
<code>\floatpagefraction</code>	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. 2298 <code>\newcommand\floatpagefraction{.5}</code>
<code>\c@dbltopnumber</code>	This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page. 2299 <code>\newcount\c@dbltopnumber</code> 2300 <code>\setcounter{dbltopnumber}{2}</code>
<code>\dbltopfraction</code>	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. 2301 <code>\newcommand\dbltopfraction{.7}</code>
<code>\dblfloatpagefraction</code>	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. 2302 <code>\newcommand\dblfloatpagefraction{.5}</code>

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 are used for both one-column mode and single-column floats in two-column mode. 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.

```
2303 \newskip\floatsep
2304 \newskip\textfloatsep
2305 \newskip\intextsep
2306 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
2307 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
2308 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

`\dblfloatsep` 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.

```
2309 \newskip\dblfloatsep
2310 \newskip\dbltextfloatsep
2311 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
2312 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

Floats on their own page or column

`\@fptop` 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.

```
2313 \newskip\@fptop
2314 \newskip\@fpsep
2315 \newskip\@fpbot
2316 \setlength\@fptop{0\p@ \@plus 1fil}
2317 \setlength\@fpsep{8\p@ \@plus 2fil}
2318 \setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblftop` Double-column ‘float pages’ in two-column mode use similar parameters.
`\@dblfpsep`
`\@dblfpbot`

```

2319 \newskip\@dblftop
2320 \newskip\@dblfpsep
2321 \newskip\@dblfpbot
2322 \setlength\@dblftop{0\p@ \@plus 1fil}
2323 \setlength\@dblfpsep{8\p@ \@plus 2fil}
2324 \setlength\@dblfpbot{0\p@ \@plus 1fil}

\topfigrule The macros can be used to put in rules between floats and text; whatever they
\botfigrule insert should be vertical mode material which takes up zero space.
\dblfigrule 2325 \let\topfigrule=\relax
2326 \let\botfigrule=\relax
2327 \let\dblfigrule=\relax
2328 \endkernel

```

File N

ltclass.dtx

77 Introduction

This file implements the following declarations, which replace `\documentstyle` in \LaTeX 2_ε 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 \LaTeX 2_ε and \LaTeX 2.09 is that \LaTeX 2_ε 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.

78 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 \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 \LaTeX 2.09 compatibility mode. As far as most packages are concerned, this only affects the warnings and errors \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 can also be called with an optional argument which is used to alter some of its behavior: option `force` or `overwrite` will allow for overwriting existing files, option `nosearch` will only check the current directory when looking if the file exists. This can be useful if you want to generate a local (modified) copy of some file that is already in the search tree of \TeX . Finally, you can use `nopreamble` to prevent it from writing the standard blurb at the top of the file (this is actually the same as using the star form of the environment).

The environment is now allowed anywhere in the document, but to ensure that all packages or options necessary are available when the document is run, it is normally best to place it at the top of your file (before `\documentclass`). A possible use case for using it inside the document body is if you want to reuse some text several times in the document you could then write it and later use `\input` to retrieve it where needed.

The begin and end tags should each be on a line by itself.

78.1 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.

79 Class and Package interface

79.1 Class name and version

`\ProvidesClass` A class can identify itself with the `\ProvidesClass{<name>}[<version>]` command. The `<version>` should begin with a date in the format `YYYY/MM/DD`.

79.2 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`.

79.3 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.

<pre>\@ifpackageloaded \@ifclassloaded \@ifpackagelater \@ifclasslater \@ifpackagewith \@ifclasswith</pre>	<p>To find out if a package has already been loaded, use</p> <pre>\@ifpackageloaded{<package>}{<true>}{<false>}.</pre> <p>To find out if a package has already been loaded with a version equal to or more recent than <i><version></i>, use</p> <pre>\@ifpackagelater{<package>}{<version>}{<true>}{<false>}.</pre> <p>To find out if a package has already been loaded with at least the options <i><options></i>, use <code>\@ifpackagewith{<package>}{<options>}{<true>}{<false>}</code>.</p>
--	---

There exists one package that can't be tested with the above commands: the `fontenc` package pretends that it was never loaded to allow for repeated reloading with different options (see `ltoutenc.dtx` for details).

79.4 Declaring new options

Options for classes and packages are built using the same macros.

<pre>\DeclareOption \DeclareOption*</pre>	<p>To define a builtin option, use <code>\DeclareOption{<name>}{<code>}</code>.</p> <p>To define the default action to perform for local options which have not been declared, use <code>\DeclareOption*{<code>}</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.

79.5 Safe Input Macros

<code>\InputIfFileExists</code>	<code>\InputIfFileExists{<file>}{<then>}{<else>}</code> Inputs <i><file></i> if it exists. Immediately before the input, <i><then></i> is executed. Otherwise <i><else></i> is executed.
<code>\IfFileExists</code>	As above, but does not input the file. One thing you might like to put in the <i><else></i> clause is
<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 ^A T _E 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.

80 Implementation

	1 (*2kernel)
<code>\if@compatibility</code>	The flag for compatibility mode. 2 \newif\if@compatibility
<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 \def\@documentclasshook{% 4 \ifx\@normalsize\undefined 5 \let\@normalsize\normalsize 6 \fi 7 }
<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@<option></code> commands are executed. All local <i><option></i> 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 \let\@declaredoptions\@empty
<code>\@classoptionslist</code>	List of options of the main class. 9 \let\@classoptionslist\relax 10 \onlypreamble\@classoptionslist

`\@unusedoptionlist` List of options of the main class that haven't been declared or loaded as class option files.

```

11 \let\@unusedoptionlist\@empty
12 \@onlypreamble\@unusedoptionlist

```

`\CurrentOption` Name of current package or option.

```

13 \let\CurrentOption\@empty

```

`\@currname` Name of current package or option.

```

14 \let\@currname\@empty

```

`\@currentx` The current file extension.

```

15 \global\let\@currentx=\@empty

```

`\@clsextension` The two possible values of `\@currentx`.

`\@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     {\@currentx}%
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\@currentx{#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
59 </2ekernel>

This internal macro is also used in \NeedsTeXFormat.

60 <latexrelease>\IncludeInRelease{2018/04/01}%
61 <latexrelease> \ifl@t@r{Guard against bad input}%
62 <*2ekernel | latexrelease>
63 \def\@ifl@t@r#1#2{%
64   \ifnum\expandafter\@parse@version@#1//00\@nil<%
65     \expandafter\@parse@version@#2//00\@nil
66     \expandafter\@secondoftwo
67   \else
68     \expandafter\@firstoftwo
69   \fi}
70 \def\@parse@version@#1{\@parse@version0#1}
71 </2ekernel | latexrelease>
72 <latexrelease>\EndIncludeInRelease
73 <latexrelease>\IncludeInRelease{0000/00/00}%
74 <latexrelease> \ifl@t@r{Guard against bad input}%
75 <latexrelease>\def\@ifl@t@r#1#2{%
76 <latexrelease> \ifnum\expandafter\@parse@version#1//00\@nil<%
77 <latexrelease> \expandafter\@parse@version#2//00\@nil
78 <latexrelease> \expandafter\@secondoftwo
79 <latexrelease> \else
80 <latexrelease> \expandafter\@firstoftwo
81 <latexrelease> \fi}
82 <latexrelease>\let\@parse@version@\undefined
83 <latexrelease>\EndIncludeInRelease
84 <*2ekernel>

85 \@onlypreamble\@ifl@t@r

86 </2ekernel>
87 <*2ekernel | latexreleasefirst>
88 \def\@parse@version#1/#2/#3#4#5\@nil{%
89 \@parse@version@dash#1-#2-#3#4\@nil
90 }

```

The `\if` test here ensures that an argument with no `/` or `-` produces 0 (actually 00).

```

91 \def\@parse@version@dash#1-#2-#3#4#5\@nil{%
92   \if\relax#2\relax\else#1\fi#2#3#4 }
93 \</2ekernel | latexreleasefirst>
94 \<*2ekernel>

```

`\@ifpackagewith` `\@ifclasswith` `\@ifpackagewith{<name>}{<option-list>}` Checks that `<option-list>` is a subset of the options **with** which `<name>` was loaded.

```

95 \def\@ifpackagewith{\@ifoptions\@pkgextension}
96 \def\@ifclasswith{\@ifoptions\@clsextension}
97 \@onlypreamble\@ifpackagewith
98 \@onlypreamble\@ifclasswith

99 \def\@ifoptions#1#2{%
100   \@expandtwoargs\@if@pti@ns{\@optionlist{#2.#1}}
101 \@onlypreamble\@ifoptions

```

Probably shouldn't use `\CurrentOption` here... (changed to `\reserved@b`.)

```

102 \</2ekernel>
103 \<latexrelease>\IncludeInRelease{2017/01/01}%
104 \<latexrelease>           {\@if@pti@ns}{Spaces in option clash check}%
105 \<*2ekernel | latexrelease>
106 \def\@if@pti@ns#1#2{%
107   \let\reserved@a\@firstoftwo

108   \edef\reserved@b{\zap@space#2 \@empty}%
109   \@for\reserved@b:=\reserved@b\do{%
110     \ifx\reserved@b\@empty
111       \else
112         \expandafter\in@\expandafter{\expandafter,\reserved@b,}{, #1,}%
113         \ifin@
114         \else
115           \let\reserved@a\@secondoftwo
116         \fi
117       \fi
118     }%
119   \reserved@a}
120 \</2ekernel | latexrelease>
121 \<latexrelease>\EndIncludeInRelease
122 \<latexrelease>\IncludeInRelease{0000/00/00}%
123 \<latexrelease>           {\@if@pti@ns}{Spaces in option clash check}%
124 \<latexrelease>\def\@if@pti@ns#1#2{%
125 \<latexrelease> \let\reserved@a\@firstoftwo
126 \<latexrelease> \@for\reserved@b:=#2\do{%
127 \<latexrelease> \ifx\reserved@b\@empty
128 \<latexrelease> \else
129 \<latexrelease> \expandafter\in@\expandafter
130 \<latexrelease>           {\expandafter,\reserved@b,}{, #1,}%
131 \<latexrelease> \ifin@
132 \<latexrelease> \else
133 \<latexrelease> \let\reserved@a\@secondoftwo
134 \<latexrelease> \fi
135 \<latexrelease> \fi
136 \<latexrelease> }%

```

```

137 <latexrelease> \reserved@a}
138 <latexrelease>\EndIncludeInRelease
139 <*2ekernel>

140 \<onlypreamble>\if@pti@ns

\ProvidesPackage Checks that the current filename is correct, and defines \ver@filename.
141 \def\ProvidesPackage#1{%
142   \xdef\@gtempa{#1}%
143   \ifx\@gtempa\@currname\else
144     \@latex@warning@no@line{You have requested
145       \<cls@pkg>\space'\@currname',\MessageBreak
146       but the \<cls@pkg>\space provides '#1'}%
147   \fi
148   \@ifnextchar[\<pr@videpackage>\<pr@videpackage>[]{}%]
149 \<onlypreamble>\ProvidesPackage

\@pr@videpackage This is the helper command for \ProvidesPackage. It tries to be cautious when
handling the identification string in case it contains UTF-8 characters.
150 </2ekernel>
151 <*2ekernel | latexrelease>
152 <latexrelease>\IncludeInRelease{2020/02/02}%
153 <latexrelease>           {\@pr@videpackage}{Protection for package info}%
154 \def\@pr@videpackage[#1]{%
155   \expandafter\protected@xdef           %      <-- protected...
156     \csname ver@\@currname.\@currentx\endcsname{#1}%
157   \ifx\@currentx\@clsextension
158     \typeout{Document Class: \<gtempa>\space#1}%
159   \else
160     \protected@wlog{Package: \<gtempa>\space#1}%  <--- protected
161   \fi}

\protected@wlog This is like plain TEX's \wlog but gracefully handles protected commands.
162 \long\def\protected@wlog#1{\begingroup
163   \set@display@protect
164   \immediate \write \m@ne {#1}\endgroup }

165 </2ekernel | latexrelease>
166 <latexrelease>\EndIncludeInRelease
167 <latexrelease>\IncludeInRelease{0000/00/00}%
168 <latexrelease>           {\@pr@videpackage}{Protection for package info}%
169 <latexrelease>
170 <latexrelease>\def\@pr@videpackage[#1]{%
171 <latexrelease>   \expandafter\xdef\csname ver@\@currname.\@currentx\endcsname{#1}%
172 <latexrelease>   \ifx\@currentx\@clsextension
173 <latexrelease>     \typeout{Document Class: \<gtempa>\space#1}%
174 <latexrelease>   \else
175 <latexrelease>     \wlog{Package: \<gtempa>\space#1}%
176 <latexrelease>   \fi}
177 <latexrelease>\let\protected@wlog\@undefined
178 <latexrelease>
179 <latexrelease>\EndIncludeInRelease
180 <*2ekernel>

181 \<onlypreamble>\@pr@videpackage

```

`\ProvidesClass` Like `\ProvidesPackage`, but for classes.
182 `\let\ProvidesClass\ProvidesPackage`
183 `\@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`

```

184 \def\ProvidesFile#1{%
185   \begingroup
186   \catcode'\ 10 %
187   \ifnum \endlinechar<256 %
188     \ifnum \endlinechar>\m@ne
189       \catcode\endlinechar 10 %
190     \fi
191   \fi
192   \@makeother\/%
193   \@makeother\&%
194   \kernel@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

During initex 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}

```

`\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.

```

195 \def\@pass@options#1#2#3{%
196   \expandafter\xdef\csname opt@#3.#1\endcsname{%
197     \@ifundefined{opt@#3.#1}\@empty
198     {\csname opt@#3.#1\endcsname},}%
199   \zap@space#2 \@empty}}
200 \@onlypreamble\@pass@options
201 \def\PassOptionsToPackage{\@pass@options\@pkgextension}
202 \def\PassOptionsToClass{\@pass@options\@clsextension}
203 \@onlypreamble\PassOptionsToPackage
204 \@onlypreamble\PassOptionsToClass

```

`\DeclareOption` Adds an option as a `\ds@` command, or the default `\default@ds` command.

```

\DeclareOption* 205 \def\DeclareOption{%
206   \let\@fileswith@pti@ns\@badrequireerror
207   \@ifstar\@defdefault@ds\@declareoption}
208 \long\def\@declareoption#1#2{%
209   \xdef\@declaredoptions{\@declaredoptions,#1}%
210   \toks@{#2}%
211   \expandafter\edef\csname ds@#1\endcsname{\the\toks@}}
212 \long\def\@defdefault@ds#1{%
213   \toks@{#1}%
214   \edef\default@ds{\the\toks@}}
215 \@onlypreamble\DeclareOption
216 \@onlypreamble\@declareoption
217 \@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.

```
218 \def\OptionNotUsed{%
219   \ifx\@current\@clsextension
220     \xdef\@unusedoptionlist{%
221       \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
222       \CurrentOption}%
223   \fi}
224 \@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*`.

```
225 % \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.

```
226 \def\ProcessOptions{%
227   \let\ds@\@empty
228   \edef\@curroptions{\@optionlist{\@currname.\@current}}%
229   \@ifstar\@xprocess@options\@process@ptions}
230 \@onlypreamble\ProcessOptions

231 \def\@process@ptions{%
232   \@for\CurrentOption:=\@declaredoptions\do{%
233     \ifx\CurrentOption\@empty\else
234       \@expandtwoargs\in@{\, \CurrentOption,}%
235       ,\ifx\@current\@clsextension\else\@classoptionslist,\fi
236       \@curroptions,}%
237     \ifin@
238       \@use@option
239       \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
240     \fi
241   \fi}%
242 \@process@ptions}
243 \@onlypreamble\@process@ptions

244 \def\@xprocess@ptions{%
245   \ifx\@current\@clsextension\else
246     \@for\CurrentOption:=\@classoptionslist\do{%
247       \ifx\CurrentOption\@empty\else
248         \@expandtwoargs\in@{\, \CurrentOption,}{\, \@declaredoptions,}%
249       \ifin@
250         \@use@option
251         \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
252       \fi
253     \fi}%
254   \fi}%
255 }
```

```

254 \fi
255 \@process@ptions}
256 \@onlypreamble\@xprocess@ptions

```

The common part of \ProcessOptions and \ProcessOptions*.

```

257 \def\@process@ptions{%
258   \@for\CurrentOption:=\@curroptions\do{%
259     \ifundefined{ds@\CurrentOption}%
260       {\@use@ption
261         \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.

```

262     \@use@ption}%

```

Clear all the definitions for option code. First set all the declared options to \relax, then reset the ‘default’ and ‘empty’ options. and the lst of declared options.

```

263   \@for\CurrentOption:=\@declaredoptions\do{%
264     \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%
265   \let\CurrentOption\@empty
266   \let\@fileswith@ptions\@fileswith@ptions
267   \AtEndOfPackage{\let\@unprocessedoptions\relax}}
268 \@onlypreamble\@process@ptions

```

\@options \@options is a synonym for \ProcessOptions* for upward compatibility with L^AT_EX2.09 style files.

```

269 \def\@options{\ProcessOptions*}
270 \@onlypreamble\@options

```

\@use@ption Execute the code for the current option.

```

271 \def\@use@ption{%
272   \@expandtwoargs\@removeelement\CurrentOption
273   \@unusedoptionlist\@unusedoptionlist
274   \csname ds@\CurrentOption\endcsname}
275 \@onlypreamble\@use@ption

```

\ExecuteOptions \ExecuteOptions{*option-list*} executes the code declared for each option.

```

276 \if2ekernel
277 \iflatexrelease\IncludeInRelease{2017/01/01}%
278 \iflatexrelease\ExecuteOptions{Spaces in \ExecuteOptions}%
279 \if*2ekernel\iflatexrelease
280 \def\ExecuteOptions#1{%

```

Use \@fortmp here as it is anyway cleared during \@for loop so does not change any existing names.

```

281 \edef\@fortmp{\zap@space#1 \@empty}%
282 \def\reserved@a##1\@nil{%
283   \@for\CurrentOption:=\@fortmp\do
284     {\csname ds@\CurrentOption\endcsname}%
285   \edef\CurrentOption{##1}}%
286 \expandafter\reserved@a\CurrentOption\@nil}

```

```

287 </2ekernel | latexrelease>
288 <latexrelease>\EndIncludeInRelease
289 <latexrelease>\IncludeInRelease{0000/00/00}%
290 <latexrelease>          {\ExecuteOptions}{Spaces in \ExecuteOptions}%
291 <latexrelease>\def\ExecuteOptions#1{%
292 <latexrelease>  \def\reserved@a##1\@nil{%
293 <latexrelease>    \for\CurrentOption:=#1\do
294 <latexrelease>      {\csname ds@\CurrentOption\endcsname}%
295 <latexrelease>    \edef\CurrentOption{##1}}%
296 <latexrelease>  \expandafter\reserved@a\CurrentOption\@nil}
297 <latexrelease>\EndIncludeInRelease
298 <*2ekernel>
299 \@onlypreamble\ExecuteOptions

```

The top-level commands, which just set some parameters then call the internal command, \@fileswithoptions.

\documentclass The main new-style class declaration.

```

300 \def\documentclass{%
301   \let\documentclass\@twoclasseserror
302   \if@compatibility\else\let\usepackage\RequirePackage\fi
303   \@fileswithoptions\@clsextension}
304 \@onlypreamble\documentclass

```

\documentstyle 2.09 style class ‘style’ declaration.

```

305 \def\documentstyle{%
306   \makeatletter\input{latex209.def}\makeatother
307   \documentclass}
308 \@onlypreamble\documentstyle

```

\RequirePackage Load package if not already loaded.

```

309 \def\RequirePackage{%
310   \@fileswithoptions\@pkgextension}
311 \@onlypreamble\RequirePackage

```

\LoadClass Load class.

```

312 \def\LoadClass{%
313   \ifx\@current\@pkgextension
314     \@latex@error
315       {\noexpand\LoadClass in package file}%
316       {You may only use \noexpand\LoadClass in a class file.}%
317   \fi
318   \@fileswithoptions\@clsextension}
319 \@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.

```

320 \def\@loadwithoptions#1#2#3{%
321   \expandafter\let\csname opt@#3.#1\expandafter\endcsname
322     \csname opt@\@currname.\@current\endcsname
323   #2{#3}}
324 \@onlypreamble\@loadwithoptions

```



```

\LoadClassWithOptions Load class ‘#1’ with the current option list.
325 \def\LoadClassWithOptions{%
326   \@loadwithoptions\@clsextension\LoadClass}
327 \@onlypreamble\LoadClassWithOptions

\RequirePackageWithOptions Load package ‘#1’ with the current option list.
328 \def\RequirePackageWithOptions{%
329   \@AtEndOfPackage{\let\@unprocessedoptions\relax}%
330   \@loadwithoptions\@pkgextension\RequirePackage}
331 \@onlypreamble\RequirePackageWithOptions

\usepackage To begin with, \usepackage produces an error. This is reset by \documentclass.

332 \def\usepackage#1{%
333   \@latex@error
334     {\noexpand \usepackage before \string\documentclass}%
335     {\noexpand \usepackage may only appear in the document
336       preamble, i.e.,\MessageBreak
337       between \noexpand\documentclass and
338       \string\begin{document}.}%
339   \@gobble}
340 \@onlypreamble\usepackage

\NeedsTeXFormat Check that the document is running on the correct system.
341 \def\NeedsTeXFormat#1{%
342   \def\reserved@a{#1}%
343   \ifx\reserved@a\fmtname
344     \expandafter\@needsformat
345   \else
346     \@latex@error{This file needs format ‘\reserved@a’%
347       \MessageBreak but this is ‘\fmtname’}{%
348       The current input file will not be processed
349       further,\MessageBreak
350       because it was written for some other flavor of
351       TeX.\MessageBreak\@ehd}%

If the file is not meant to be processed by LATEX 2ε we stop inputting it, but we
do not end the run. We just end inputting the current file.
352   \endinput \fi}
353 \@onlypreamble\NeedsTeXFormat

354 \def\@needsformat{%
355   \@ifnextchar[%]
356     \@needsformat
357   {}}
358 \@onlypreamble\@needsformat

359 \def\@needsformat[#1]{%
360   \ifl@t@r\fmtversion{#1}{}%
361   {\@latex@warning@no@line
362     {You have requested release ‘#1’ of LaTeX,\MessageBreak
363     but only release ‘\fmtversion’ is available}}}
364 \@onlypreamble\@needsformat

```

`\zap@space` `\zap@space foo<space>\@empty` removes all spaces from `foo` that are not protected by `{ }` groups.

```
365 \def\zap@space#1 #2{%
366   #1%
367   \ifx#2\@empty\else\expandafter\zap@space\fi
368   #2}
```

`\@fileswithoptions` The common part of `\documentclass` and `\usepackage`.

```
369 \def\@fileswithoptions#1{%
370   \@ifnextchar[%]
371     {\@fileswith@ptions#1}%
372     {\@fileswith@ptions#1[]}}
373 \@onlypreamble\@fileswithoptions

374 \def\@fileswith@ptions#1[#2]#3{%
375   \@ifnextchar[%]
376     {\@fileswith@ptions#1[#{#2}]#3}%
377     {\@fileswith@ptions#1[#{#2}]#3[]}}
378 \@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.

```
379 (/2ekernel)
380 \latexrelease\IncludeInRelease{2017/01/01}%
381 \latexrelease) {\@fileswith@ptions}{\ifx tests in \@fileswith@ptions}%
382 (*2ekernel|latexrelease)
383 \def\@fileswith@ptions#1[#2]#3[#4]{%
384   \ifx#1\@clsextension
385     \ifx\@classoptionslist\relax
386       \xdef\@classoptionslist{\zap@space#2 \@empty}%
387       \def\reserved@a{%
388         \@onefilewithoptions#3[#{#2}][#{#4}]#1%
389         \@documentclasshook}%
390     \else
391       \def\reserved@a{%
392         \@onefilewithoptions#3[#{#2}][#{#4}]#1%
393       \fi
394     \else
```

build up a list of calls to `\@onefilewithoptions` (one for each package) without thrashing the parameter stack.

```
395   \def\reserved@b##1,{%
```

If `#1` is `\@nnil` we have reached the end of the list (older version used `\@nil` here but `\@nil` is undefined so `\ifx` equal to all undefined commands)

```
396     \ifx\@nnil##1\relax\else
```

If `\ifx\@nnil##1\n@nil` is true then #1 is (presumably) empty (Older code used `\relax` which is slightly easier to get into #1 by mistake, which would spoil this test.)

```

397         \ifx\@nnil##1\@nnil\else
398             \noexpand\@onefilewithoptions##1[#{2}][#{4}]%
399             \noexpand\@pkgextension
400         \fi
401         \expandafter\reserved@b
402     \fi}%
403     \edef\reserved@a{\zap@space#3 \@empty}%
404     \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nnil,}%
405 \fi
406 \reserved@a}
407 </2ekernel | latexrelease>

408 <latexrelease>\EndIncludeInRelease
409 <latexrelease>\IncludeInRelease{0000/00/00}%
410 <latexrelease>         {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
411 <latexrelease>\def\@fileswith@pti@ns#1[#2]#3[#4]{%
412 <latexrelease> \ifx#1\@clsextension
413 <latexrelease>     \ifx\@classoptionslist\relax
414 <latexrelease>         \xdef\@classoptionslist{\zap@space#2 \@empty}%
415 <latexrelease>         \def\reserved@a{%
416 <latexrelease>             \@onefilewithoptions#3[#{2}][#{4}]#1%
417 <latexrelease>             \@documentclasshook}%
418 <latexrelease>     \else
419 <latexrelease>         \def\reserved@a{%
420 <latexrelease>             \@onefilewithoptions#3[#{2}][#{4}]#1}%
421 <latexrelease>     \fi
422 <latexrelease> \else
423 <latexrelease>     \def\reserved@b##1,{%
424 <latexrelease>         \ifx\@nil##1\relax\else
425 <latexrelease>         \ifx\relax##1\relax\else
426 <latexrelease>             \noexpand\@onefilewithoptions##1[#{2}][#{4}]%
427 <latexrelease>             \noexpand\@pkgextension
428 <latexrelease>         \fi
429 <latexrelease>         \expandafter\reserved@b
430 <latexrelease>     \fi}%
431 <latexrelease>     \edef\reserved@a{\zap@space#3 \@empty}%
432 <latexrelease>     \edef\reserved@a{%
433 <latexrelease>         \expandafter\reserved@b\reserved@a,\@nil,}%
434 <latexrelease> \fi
435 <latexrelease> \reserved@a}
436 <latexrelease>\EndIncludeInRelease
437 <*2ekernel>

438 \@onlypreamble\@fileswith@pti@ns

    Have the main argument as #1, so we only need one \expandafter above.
439 \def\@onefilewithoptions#1[#2][#3]#4{%
440     \pushfilename
441     \xdef\@currname{#1}%
442     \global\let\@current#4%
443     \let\CurrentOption\@empty
444     \@reset@ptions

```

445 \makeatletter

Grab everything in a macro, so the parameter stack is popped before any processing begins.

```

446 \def\reserved@a{%
447   \ifl@aded\@currentx{#1}%
448   {\if@options\@currentx{#1}{#2}{}%
449     {\@latex@error
450       {Option clash for \cls@pkg\space #1}%
451       {The package #1 has already been loaded
452         with options:\MessageBreak
453         \space\space[\@optionlist{#1.\@currentx}]\MessageBreak
454         There has now been an attempt to load it
455         with options:\MessageBreak
456         \space\space[#2]\MessageBreak
457         Adding the global options:\MessageBreak
458         \space\space
459         \@optionlist{#1.\@currentx},#2\MessageBreak
460         to your \noexpand\documentclass declaration may fix this.%
461         \MessageBreak
462         Try typing \space <return> \space to proceed.}}}%
463   {\@pass@options\@currentx{#2}{#1}%

464   \global\expandafter
465   \let\csname ver@\@currname.\@currentx\endcsname\@empty

```

We initialize \dots-h@@k here and only if we load the file so that it remains undefined otherwise.

```

466   \expandafter\let\csname\@currname.\@currentx-h@@k\endcsname\@empty
467   \InputIfFileExists
468     {\@currname.\@currentx}%
469     {}%
470     {\@missingfileerror\@currname\@currentx}%

```

\@unprocessedoptions will generate an error for each specified option in a package unless a \ProcessOptions has appeared in the package file.

```

471   \let\@unprocessedoptions\@unprocessedoptions
472   \csname\@currname.\@currentx-h@@k\endcsname
473   \expandafter\let\csname\@currname.\@currentx-h@@k\endcsname
474     \@undefined
475   \@unprocessedoptions}%

476   \@ifl@ter\@currentx{#1}{#3}{}%
477   {\@latex@warning@no@line
478     {You have requested,\on@line,
479     version\MessageBreak
480     ‘#3’ of \cls@pkg\space #1,\MessageBreak
481     but only version\MessageBreak
482     ‘\csname ver@#1.\@currentx\endcsname’\MessageBreak
483     is available}}}%

484   \ifx\@currentx\cls@extension\let\LoadClass\@twoloadclasserror\fi
485   \popfilename
486   \@reset@options}%
487   \reserved@a}
488 \@onlypreamble\@onefilewithoptions

```

`\@fileswith@pti@ns` Save the definition (for error checking).
489 `\let\@fileswith@pti@ns\@fileswith@pti@ns`
490 `\@onlypreamble\@fileswith@pti@ns`

`\@reset@ptions` Reset the default option, and clear lists of declared options.
491 `\def\@reset@ptions{%`
492 `\global\ifx\@currentx\@clsextension`
493 `\let\default@ds\OptionNotUsed`
494 `\else`
495 `\let\default@ds\@unknownoptionerror`
496 `\fi`
497 `\global\let\ds@\@empty`
498 `\global\let\@declaredoptions\@empty}`
499 `\@onlypreamble\@reset@ptions`

80.1 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` 500 `\ifx\@begindocumenthook\@undefined`
501 `\let\@begindocumenthook\@empty`
502 `\fi`
503 `\let\@enddocumenthook\@empty`

`\g@addto@macro` Globally add to the end of a macro.
504 `\long\def\g@addto@macro#1#2{%`
505 `\begingroup`
506 `\toks@\expandafter{#1#2}%`
507 `\xdef#1{\the\toks@}%`
508 `\endgroup}`

`\AtEndOfPackage` The access functions.
`\AtEndOfClass` 509 `\def\AtEndOfPackage{%`
`\AtBeginDocument` 510 `\expandafter\g@addto@macro\csname\@currname.\@currentx-h@@k\endcsname}`
`\AtEndDocument` 511 `\let\AtEndOfClass\AtEndOfPackage`
512 `\@onlypreamble\AtEndOfPackage`
513 `\@onlypreamble\AtEndOfClass`
514 `\DeclareRobustCommand\AtBeginDocument{\g@addto@macro\@begindocumenthook}`
515 `\DeclareRobustCommand\AtEndDocument{\g@addto@macro\@enddocumenthook}`
516 `\@onlypreamble\AtBeginDocument`

`\@cls@pkg` The current file type.
517 `\def\@cls@pkg{%`
518 `\ifx\@currentx\@clsextension`
519 `document class%`
520 `\else`
521 `package%`
522 `\fi}`
523 `\@onlypreamble\@cls@pkg`

```

\@unknownoptionerror Bad option.
524 \def\@unknownoptionerror{%
525   \@latex@error
526   {Unknown option '\CurrentOption' for \@cls@pkg\space'\@currname'}%
527   {The option '\CurrentOption' was not declared in
528     \@cls@pkg\space'\@currname', perhaps you\MessageBreak
529     misspelled its name.
530     Try typing \space <return>
531     \space to proceed.}}
532 \@onlypreamble\@unknownoptionerror

\@@unprocessedoptions Declare an error for each option, unless a \ProcessOptions occurred.
533 \def\@@unprocessedoptions{%
534   \ifx\@current\@pkgextension
535     \edef\@curoptions{\@optionlist{\@currname.\@current}}%
536     \@for\CurrentOption:=\@curoptions\do{%
537       \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
538   \fi}
539 \@onlypreamble\@@unprocessedoptions
540 \@onlypreamble\@@unprocessedoptions

\@badrequireerror \RequirePackage or \LoadClass occurs in the options section.
541 \def\@badrequireerror#1[#2]#3[#4]{%
542   \@latex@error
543   {\noexpand\RequirePackage or \noexpand\LoadClass
544     in Options Section}%
545   {The \@cls@pkg\space '\@currname' is defective.\MessageBreak
546     It attempts to load '#3' in the options section, i.e.,\MessageBreak
547     between \noexpand\DeclareOption and \string\ProcessOptions.}}
548 \@onlypreamble\@badrequireerror

\@twoloadclasserror Two \LoadClass in a class.
549 \def\@twoloadclasserror{%
550   \@latex@error
551   {Two \noexpand\LoadClass commands}%
552   {You may only use one \noexpand\LoadClass in a class file}}
553 \@onlypreamble\@twoloadclasserror

\@twoclasseserror Two \documentclass or \documentstyle.
554 \def\@twoclasseserror#1#{%
555   \@latex@error
556   {Two \noexpand\documentclass or \noexpand\documentstyle commands}%
557   {The document may only declare one class.}\@gobble}
558 \@onlypreamble\@twoclasseserror

```

80.2 Providing shipment

```

\two@digits Prefix a number less than 10 with '0'.
559 \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.

```

```

560 % \changes{v1.3a}{2019/07/01}{Support UTF8 and spaces in
561 %                               filecontents environment file name}
562 % \changes{v1.3b}{2019/08/27}{Make various commands robust}
563 % \changes{v1.3c}{2019/09/11}{Support optional argument for filecontents}
564 % \changes{v1.3f}{2020/01/05}{Support more write streams in LuaTeX gh/238}
565 %
566 </2ekernel>
567 <*2ekernel | latexrelease>
568 <latexrelease>\IncludeInRelease{2019/10/01}%
569 <latexrelease>                               {\filec@ntents}{Spaces in file names + optional arg}%
570 %

```

We use @tempswa to mean no preamble writing and reuse @filesw to indicate no overwriting:

```

571 \def\filecontents{\@tempswatrue\@fileswtrue
572   \ifnextchar[\filec@ntents@opt\filec@ntents
573 }
574 \@namedef{filecontents*}{\@tempswafalse\@fileswtrue
575   \ifnextchar[\filec@ntents@opt\filec@ntents
576 }

```

To handle the optional argument we execute for each option the command \filec@ntents@OPTION if it exist or complain about unknown option.

```

577 \def\filec@ntents@opt[#1]{%
578   \for\@tempa:=#1\do{%
579     \ifcsname filec@ntents@\@tempa\endcsname
580       \csname filec@ntents@\@tempa\endcsname
581     \else
582       \@latexerror{Unknown filecontents option #1}%
583       {Valid options are force (or overwrite), nosearch, noheader}%
584     \fi}%
585   \filec@ntents
586 }

```

Option force) (or overwrite) changes the overwriting switch

```

587 \let\filec@ntents@force\@fileswfalse
588 \let\filec@ntents@overwrite\@fileswfalse % alternative name

```

and option noheader the preamble switch (which is equivalent to using the star form of the environment).

```

589 \let\filec@ntents@noheader\@tempswafalse

```

Option nosearch only checks the current directory not the how T_EX tree for the existence of the file to write.

```

590 \def\filec@ntents@nosearch{%
591   \let\filec@ntents@checkdir\@currdir
592   \def\filec@ntents@where{in current directory}}

```

By default we search the whole tree:

```

593 \let\filec@ntents@checkdir\@empty
594 \def\filec@ntents@where{exists on the system}

595 \begingroup%
596 \@tempcnta=1
597 \loop
598   \catcode\@tempcnta=12 %

```

```

599 \advance\@tempcnta\@ne %
600 \ifnum\@tempcnta<32 %
601 \repeat %
602 \catcode'\*=11 %
603 \catcode'\^M\active%
604 \catcode'\^L\active\let^^L\relax%
605 \catcode'\^I\active%

606 \gdef\filec@ntents#1{%
607 \set@curr@file{\filec@ntents@checkdir#1}%
608 \edef\q@curr@file{\expandafter\quote@name\expandafter{\@curr@file}}%

    LuaTeX has more writes (and 18 is safe here).

609 \chardef\reserved@c\ifx\directlua\@undefined 15 \else 127 \fi%
610 \openin\@inputcheck\q@curr@file \space %
611 \ifeof\@inputcheck%
612 \@latex@warning@no@line%
613 {Writing file '\@currdir\@curr@file'}%

614 \ch@ck7\reserved@c\write\relax%
615 \immediate\openout\reserved@c\q@curr@file\relax%
616 \else%

617 \if@filesw%
618 \@latex@warning@no@line%
619 {File '\@curr@file' already \filec@ntents@where.\MessageBreak%
620 Not generating it from this source}%
621 \let\write\@gobbles%
622 \let\closeout\@gobble%
623 \else%

```

If we are overwriting, we try to make sure that the user is not by mistake overwriting the input file (`\jobname`). Of course, this only works for input files ending in `.tex`. If a different extension is used there is no way to see that we are overwriting ourselves!

```

624 \edef\reserved@a{#1}%
625 \edef\reserved@a{\detokenize\expandafter{\reserved@a}}%
626 \edef\reserved@b{\detokenize\expandafter{\jobname}}%
627 \ifx\reserved@a\reserved@b%
628 \@fileswtrue%
629 \else%
630 \edef\reserved@b{\reserved@b\detokenize{.tex}}%
631 \ifx\reserved@a\reserved@b
632 \@fileswtrue%
633 \fi%
634 \fi%

```

We allocate a write channel but we open it only if it is (hopefully) safe. If not opened that means we are going to write on the terminal.

```

635 \ch@ck7\reserved@c\write\relax%
636 \if@filesw% % Foul ... trying to overwrite \jobname!
637 \@latex@error{Trying to overwrite '\jobname.tex'}{You can't %
638 write to the file you are reading from!\MessageBreak%
639 Data is written to screen instead.}%
640 \else%
641 \@latex@warning@no@line%

```



```

642         {Writing or overwriting file '@currdir\@curr@file'}%
643         \immediate\openout\reserved@c#1\relax%
644     \fi%
645 \fi%
646 \fi%

```

Closing the \@inputcheck is done here to avoid having to do this in each branch.

```

647 \closein\@inputcheck%
648 \if@tempwa%

649     \immediate\write\reserved@c{%
650         \@percentchar\@percentchar\space%
651         \expandafter\@gobble\string\LaTeX2e file '@curr@file'^^J%
652         \@percentchar\@percentchar\space generated by the %
653         '@currentvir' \expandafter\@gobblefour\string\newenvironment^^J%
654         \@percentchar\@percentchar\space from source 'jobname' on %
655         \number\year/\two@digits\month/\two@digits\day.^^J%
656         \@percentchar\@percentchar}%
657 \fi%
658 \let\do\@makeother\dospecials%

```

If there are active characters in the upper half (e.g., from `inputenc` there would be confusion so we render everything harmless.

```

659 \count@ 128\relax%
660 \loop%
661     \catcode\count@ 11\relax%
662     \advance\count@ \@ne%
663     \ifnum\count@<\@ccclvi%
664 \repeat%

665 \edef\E{\@backslashchar end\string{\@currentvir\string}}%
666 \edef\reserved@b{%
667     \def\noexpand\reserved@b%
668         #####1\E####2\E####3\relax}%
669 \reserved@b{%
670     \ifx\relax##3\relax%

```

There was no \end{filecontents}

```

671     \immediate\write\reserved@c{##1}%
672 \else%

```

There was a \end{filecontents}, so stop this time.

```

673     \edef^^M{\noexpand\end{\@currentvir}}%
674     \ifx\relax##1\relax%
675 \else%

```

Text before the \end, write it with a warning.

```

676     \@latex@warning{Writing text '##1' before %
677         \string\end{\@currentvir}\MessageBreak as last line of \@curr@file}%
678     \immediate\write\reserved@c{##1}%
679 \fi%
680 \ifx\relax##2\relax%
681 \else%

```

Text after the \end, ignore it with a warning.

```

682     \@latex@warning{%
683         Ignoring text '##2' after \string\end{\@currentvir}}%

```

```

684     \fi%
685     \fi%
686     ^^M}%

687 \catcode'\^^L\active%
688 \let\L\@undefined%
689 \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
690 \catcode'\^^I\active%
691 \let\I\@undefined%
692 \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
693 \catcode'\^^M\active%
694 \edef^^M##1^^M{%
695     \noexpand\reserved@b##1\E\E\relax}}%
696 \endgroup%

697 \</2ekernel | latexrelease>
698 \<latexrelease>\EndIncludeInRelease
699 \<latexrelease>\IncludeInRelease{0000/00/00}%
700 \<latexrelease>           {\filec@ntents}{Spaces in file names + optional arg}%
701 \<latexrelease>
702 \<latexrelease>\let\filec@ntents@opt      \@undefined
703 \<latexrelease>\let\filec@ntents@force    \@undefined
704 \<latexrelease>\let\filec@ntents@overwrite \@undefined
705 \<latexrelease>\let\filec@ntents@noheader \@undefined
706 \<latexrelease>\let\filec@ntents@nosearch \@undefined
707 \<latexrelease>\let\filec@ntents@checkdir \@undefined
708 \<latexrelease>\let\filec@ntents@where    \@undefined
709 \<latexrelease>
710 \<latexrelease>\begingroup%
711 \<latexrelease>\@tempcnta=1
712 \<latexrelease>\loop
713 \<latexrelease> \catcode\@tempcnta=12 %
714 \<latexrelease> \advance\@tempcnta\@ne %
715 \<latexrelease>\ifnum\@tempcnta<32 %
716 \<latexrelease>\repeat %
717 \<latexrelease>\catcode'\*=11 %
718 \<latexrelease>\catcode'\^^M\active%
719 \<latexrelease>\catcode'\^^L\active\let^^L\relax%
720 \<latexrelease>\catcode'\^^I\active%
721 \<latexrelease>
722 \<latexrelease>\gdef\filec@ntents#1{%
723 \<latexrelease> \openin\@inputcheck#1 %
724 \<latexrelease> \ifeof\@inputcheck%
725 \<latexrelease> \@latex@warning@no@line%
726 \<latexrelease> {Writing file '\@currdir#1'}%
727 \<latexrelease> \chardef\reserved@c15 %
728 \<latexrelease> \ch@ck7\reserved@c\write%
729 \<latexrelease> \immediate\openout\reserved@c#1\relax%
730 \<latexrelease> \else%
731 \<latexrelease> \closein\@inputcheck%
732 \<latexrelease> \@latex@warning@no@line%
733 \<latexrelease> {File '#1' already exists on the system.\MessageBreak%
734 \<latexrelease> Not generating it from this source}%
735 \<latexrelease> \let\write\@gobbletwo%
736 \<latexrelease> \let\closeout\@gobble%

```

```

737 \latexrelease\fi%
738 \latexrelease\if@tempswa%
739 \latexrelease\immediate\write\reserved@c{%
740 \latexrelease\@percentchar\@percentchar\space%
741 \latexrelease\expandafter\@gobble\string\LaTeXe file '#1'^^J%
742 \latexrelease\@percentchar\@percentchar\space generated by the %
743 \latexrelease'\@currentenvir'\expandafter\@gobblefour\string\newenvironment^^J%
744 \latexrelease\@percentchar\@percentchar\space from source '\jobname' on %
745 \latexrelease\number\year/\two@digits\month/\two@digits\day.^^J%
746 \latexrelease\@percentchar\@percentchar}%
747 \latexrelease\fi%
748 \latexrelease\let\do\@makeother\dospecials%
749 \latexrelease\count@ 128\relax%
750 \latexrelease\loop%
751 \latexrelease\catcode\count@ 11\relax%
752 \latexrelease\advance\count@ \@ne%
753 \latexrelease\ifnum\count@<\@ccclvi%
754 \latexrelease\repeat%
755 \latexrelease\edef\E{\@backslashchar end\string{\@currentenvir\string}}%
756 \latexrelease\edef\reserved@b{%
757 \latexrelease\def\noexpand\reserved@b%
758 \latexrelease####1\E####2\E####3\relax}%
759 \latexrelease\reserved@b{%
760 \latexrelease\ifx\relax##3\relax%
761 \latexrelease\immediate\write\reserved@c{##1}%
762 \latexrelease\else%
763 \latexrelease\edef^^M{\noexpand\end{\@currentenvir}}%
764 \latexrelease\ifx\relax##1\relax%
765 \latexrelease\else%
766 \latexrelease\@latex@warning{Writing text '#1' before %
767 \latexrelease\string\end{\@currentenvir}\MessageBreak as last line of #1}%
768 \latexrelease\immediate\write\reserved@c{##1}%
769 \latexrelease\fi%
770 \latexrelease\ifx\relax##2\relax%
771 \latexrelease\else%
772 \latexrelease\@latex@warning{%
773 \latexreleaseIgnoring text '##2' after \string\end{\@currentenvir}}%
774 \latexrelease\fi%
775 \latexrelease\fi%
776 \latexrelease^^M}%
777 \latexrelease\catcode'\^^L\active%
778 \latexrelease\let\L\@undefined%
779 \latexrelease\def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
780 \latexrelease\catcode'\^^I\active%
781 \latexrelease\let\I\@undefined%
782 \latexrelease\def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
783 \latexrelease\catcode'\^^M\active%
784 \latexrelease\edef^^M##1^^M{%
785 \latexrelease\noexpand\reserved@b##1\E\E\relax}}%
786 \latexrelease\endgroup%
787 \latexrelease\EndIncludeInRelease
788 \latexrelease\*2ekernel)
789 \begingroup

```

```

791 \catcode'\=\catcode'\%
792 \catcode'\%=12
793 \catcode'\*=11
794 \gdef\@percentchar{%
795 \gdef\endfilecontents{
796 \immediate\closeout\reserved@
797 \def\T##1##2##3{
798 \ifx##1\@undefined\else
799 \latex@warning@no@line{##2 has been converted to Blank ##3e}|
800 \fi}|
801 \T\L{Form Feed}{Lin}|
802 \T\I{Tab}{Spac}|
803 \immediate\write\@unused{}}
804 \global\let\endfilecontents*\endfilecontents

```

We no longer prevent the code to be used after begin document (no rollback needed for this change).

```

805 %\@onlypreamble\filecontents
806 %\@onlypreamble\endfilecontents
807 %\@onlypreamble\filecontents*
808 %\@onlypreamble\endfilecontents*
809 \endgroup
810 %\@onlypreamble\filecontents

```

81 Package/class rollback mechanism

```

811 \if2kernel
812 \if*2kernel\latexreleasefirst

```

`\pkgcls@debug` For testing we have a few extra lines of code that by default do nothing but one can set `\pkgcls@debug` to `\typeout` to get extra info. Sometime in the future this will be dropped.

```

813 \if*tracerollback
814 \let\pkgcls@debug\typeout
815 \let\pkgcls@debug\@gobble
816 \endiftracerollback

```

`\requestedLaTeXdate` The macro (!) `\requestedLaTeXdate` holds the globally requested rollback date (via `\latexrelease`) or zero if no such request was made.

```

817 \def\requestedLaTeXdate{0}

```

`\pkgcls@targetdate` If a rollback for a package or class is requested then `\pkgcls@targetdate` holds the requested date as a number YYYYMMDD (if there was one, otherwise the value of `\requestedLaTeXdate`) and `\pkgcls@targetlabel` will be empty. If there was a request for a named version then `\pkgcls@targetlabel` holds the version name and `\pkgcls@targetdate` is set to 1.

`\pkgcls@targetlabel`
`\pkgcls@innerdate`

`\pkgcls@targetdate=0` is used to indicate that there was no rollback request. While loading an old release `\pkgcls@targetdate` is also reset to zero so that `\DeclareRelease` declarations are bypassed.

In contrast `\pkgcls@innerdate` will always hold the requested date (in a macro not a counter) if there was one, otherwise, e.g., if there was no request or a request

to a version name it will contain T_EX largest legal number. While loading a file this can be used to provide conditionals that select code based on the request.

```
818 \ifx\pkgcls@targetdate\@undefined
819   \newcount\pkgcls@targetdate
820 \fi
821 \let\pkgcls@targetlabel\@empty
822 \def\pkgcls@innerdate{\maxdimen}
```

`\pkgcls@candidate` When looping through the `\DeclareRelease` declarations we record if the release is the best candidate we have seen so far. This is recorded in `\pkgcls@candidate` and we update it whenever we see a better one.

In `\pkgcls@releasedate` we keep track of the release date of that candidate.

```
823 \let\pkgcls@candidate\@empty
824 \let\pkgcls@releasedate\@empty
```

`\load@onefilewithoptions` the best place to add the rollback code is at the point where `\@onefilewithoptions` is called to load a single class or package.

To make things easy we save the old definition as `\load@onefilewithoptions` and then provide a new interface.

Important: as this code is also unconditionally placed into `latexrelease` we can only do this name change once otherwise both macros will contain the same code.

```
825 \ifx\load@onefilewithoptions\@undefined
826   \let\load@onefilewithoptions\@onefilewithoptions
827   \def\@onefilewithoptions#1[#2][#3]#4{%
```

First a bit of tracing normally disabled.

```
828 (*tracerollback)
829   \pkgcls@debug{--- File loaded request (\noexpand\usepackage or ...)}%
830   \pkgcls@debug{\@spaces 1: #1}%
831   \pkgcls@debug{\@spaces 2: #2}%
832   \pkgcls@debug{\@spaces 3: #3}%
833   \pkgcls@debug{\@spaces 4: #4}%
834 </tracerollback>
```

Two of the arguments are needed later on in error/warning messages so we save them.

```
835   \def\pkgcls@name{#1}%           % for info message
836   \def\pkgcls@arg {#3}%           % for info message
```

then we parse the final optional argument to determine if there is a specific rollback request for the current file. This will set `\pkgcls@targetdate`, `\pkgcls@targetlabel` and `\pkgcls@mindate`.

```
837   \pkgcls@parse@date@arg{#3}%
```

When determining the correct release to load we keep track of candidates in `\pkgcls@candidate` and initially we don't have any:

```
838   \let\pkgcls@candidate\@empty
```

If we had a rollback request then `#3` may contain data but not necessarily a “minimal date” so instead of passing it on we pass on the content of `\pkgcls@mindate`. We need to pass the value not the command, otherwise nested packages may pick up the wrong information.

```
839   \begingroup
```

```

840 \edef\reserved@a{%
841   \endgroup
842   \unexpanded{\load@onefilewithoptions#1[#2]}%
843   [\pkgcls@mindate]%
844   \unexpanded{#4}}%
845 \reserved@a
846 }
847 \fi

```

`\pkgcls@parse@date@arg` The `\pkgcls@parse@date@arg` command parses the second optional argument of `\usepackage`, `\RequirePackage` or `\documentclass` for a rollback request setting the values of `\pkgcls@targetdate` and `\pkgcls@targetlabel`.

This optional argument has a dual purpose: If it just contains a date string then this means that the package should have at least that date (to ensure that a certain feature is actually available, or a certain bug has been fixed). When the package gets loaded the information in `\Provides...` will then be checked against this request.

But if it starts with an equal sign followed by a date string or followed by a version name then this means that we should roll back to the state of the package at the date or to the version with the requested name.

If there was no optional argument or the optional argument does not start with “=” then the `\pkgcls@targetdate` is set to the date of the overall rollback request (via `latexrelease`) or if that was not given it is set to 0. In either case `\pkgcls@targetlabel` will be made empty.

If the argument doesn’t start with “=” then it is supposed to be a “minimal date” and we therefore save the value in `\pkgcls@mindate`, otherwise this macro is made empty.

So in summary we have:

Input	<code>\pkgcls@targetdate</code>	<code>\pkgcls@targetlabel</code>	<code>\pkgcls@mindate</code>
<code><empty></code>	<code><global-rollbackdate-as-number></code>	<code><empty></code>	<code><empty></code>
<code><date></code>	<code><global-rollbackdate-as-number></code>	<code><empty></code>	<code><date></code>
<code>=<date></code>	<code><date-as-number></code>	<code><empty></code>	<code><empty></code>
<code>=<version></code>	1	<code><version></code>	<code><empty></code>
<code><other></code>	<code><global-rollbackdate-as-number></code>	<code><empty></code>	<code><other></code>

where `<global-rollbackdate-as-number>` is a date request given via `latexrelease` or if there wasn’t one 0.

```

848 \def\pkgcls@parse@date@arg #1{%

```

If the argument is empty we use the rollback date from `latexrelease` which has the value of zero if there was no rollback request. The label and the minimal date is made empty in that case.

```

849   \ifx\@nil#1\@nil
850     \pkgcls@targetdate\requestedLaTeXdate\relax
851     \let\pkgcls@targetlabel\@empty
852     \let\pkgcls@mindate\@empty

```

Otherwise we parse the argument further, checking for a = as the first character. We append a = at the end so that there is at least one such character in the argument.

```

853   \else

```

```

854     \pkgcls@parse@date@arg@#1=\@nil\relax
855     \fi
856 }

```

The actual parsing work then happens in `\pkgcls@parse@date@arg@`:

```

857 \def\pkgcls@parse@date@arg@#1=#2\@nil{%

```

We set `\pkgcls@targetdate` depending on the parsing result; the code is expandable so we can do the parsing as part of the assignment.

```

858   \pkgcls@targetdate

```

If a `=` was in first position then `#1` will be empty. In that case `#2` will be the original argument with a `=` appended.

This can be parsed with `\@parse@version`, the trailing character is simply ignored. This macro returns the parsed date as a number (or zero if it wasn't a date) and accepts both `YYYY/MM/DD` and `YYYY-MM-DD` formats.

```

859     \ifx\@nil#1\@nil
860     \@parse@version0#2//00\@nil\relax

```

Whatever is returned is thus assigned to `\pkgcls@targetdate` and therefore we can now test its value. If the value is zero we assume that the remaining argument string represents a version and change `\pkgcls@targetdate` and set `\pkgcls@targetlabel` to the version name (after stripping off the trailing `=`).

```

861     \ifnum \pkgcls@targetdate=\z@
862     \pkgcls@targetdate\@ne
863     \def\pkgcls@innerdate{\maxdimen}%
864     \pkgcls@parse@date@arg@version#2%
865     \else
866     \edef\pkgcls@innerdate{\the\pkgcls@targetdate}%
867     \fi
868     \let\pkgcls@mindate\@empty
869     \else

```

If `#1` was not empty then there wasn't a `=` character in first position so we are dealing either with a "minimum date" or with some incorrect data. We assume the former and make the following assignments (the first one finishing the assignment of `\pkgcls@targetdate`):

```

870     \requestedLaTeXdate\relax
871     \let\pkgcls@targetlabel\@empty
872     \def\pkgcls@innerdate{\maxdimen}%
873     \def\pkgcls@mindate{#1}%

```

If the min-date is after the requested rollback date (if there is any, i.e., if it is not zero) then we have a conflict and therefore issue a warning.

```

874     \ifnum \pkgcls@targetdate > \z@
875     \ifnum \@parse@version0#1//00\@nil > \pkgcls@targetdate
876     \@latex@warning@no@line{Suspicious rollback/min-date date given\MessageBreak
877     A minimal date of #1 has been specified for
878     \cls@pkg\MessageBreak '\pkgcls@name'.\MessageBreak
879     But this is in conflict
880     with a rollback request to \requestedpatchdate}
881     \fi
882     \fi
883     \fi
884 }

```

Strip off the trailing = and assign the version name to `\pkgcls@targetlabel`.

```
885 \def\pkgcls@parse@date@arg@version#1={%
886   \def\pkgcls@targetlabel{#1}}
```

\DeclareRelease First argument is the “name” of the release and it can be left empty if one doesn’t like to give a name to the release. The second argument is that from which on this release was available (or should be used in case of minor updates). The final argument is the external file name of this release, by convention this should be `<pkg/cls-name>-<date>.<extension>` but this is not enforced and through this argument one can overwrite it.

```
887 \def\DeclareRelease#1#2#3{%
888   \ifnum\pkgcls@targetdate>\z@ % some sort of rollback request
889 (*traceroollback)
890   \pkgcls@debug{---\string\DeclareRelease:}%
891   \pkgcls@debug{\@spaces 1: #1}%
892   \pkgcls@debug{\@spaces 2: #2}%
893   \pkgcls@debug{\@spaces 3: #3}%
894 (/traceroollback)
```

If the date argument #2 is empty we are dealing with a special release that should be only accessible via its name; a typical use case would be a “beta” release. So if we are currently processing a date request we ignore it and otherwise we check if we can match the name and if so load the corresponding release file.

```
895   \ifx\@nil#2\@nil
896     \ifnum\pkgcls@targetdate=\@ne % named request
897       \def\reserved@a{#1}%
898       \ifx\pkgcls@targetlabel\reserved@a
899         \pkgcls@use@this@release{#3}{}%
900 (*traceroollback)
901       \else
902         \pkgcls@debug{Label doesn't match}%
903 (/traceroollback)
904     \fi
905 (*traceroollback)
906   \else
907     \pkgcls@debug{Date request: ignored}%
908 (/traceroollback)
909   \fi
910 \else
```

If the value of `\pkgcls@targetdate` is greater than 1 (or in reality greater than something like 19930101) we are dealing with a rollback request to a specific date.

```
911   \ifnum\pkgcls@targetdate>\@ne % a real request
```

So we parse the date of this release to check if it is before or after the request date.

```
912   \ifnum\@parse@version#2//00\@nil
913     >\pkgcls@targetdate
```

If it is after we have to distinguish between two cases: If there was an earlier candidate we use that one because the other is too late, but if there wasn’t one (i.e., if current release is the oldest that exists) we use it as the best choice. However in that case something is wrong (as there shouldn’t be a rollback to a date where a package used doesn’t yet exists. So we make a complained to the user.


```

914         \ifx\pkgcls@candidate\@empty
915             \pkgcls@rollbackdate@error{#2}%
916             \pkgcls@use@this@release{#3}{#2}%
917         \else
918             \pkgcls@use@this@release\pkgcls@candidate
919                                     \pkgcls@releasedate
920         \fi
921     \else

```

Otherwise, if the release date of this version is before the target rollback and we record it as a candidate. But we don't use it yet as there may be another release which is still before the target rollback.

```

922         \def\pkgcls@candidate{#3}%
923         \def\pkgcls@releasedate{#2}%
924 (*tracerollback)
925         \pkgcls@debug{New candidate: #3}%
926 (/tracerollback)
927     \fi
928 \else

```

If we end up in this branch we have a named version request. So we check if `\pkgcls@targetlabel` matches the current name and if yes we use this release immediately, otherwise we do nothing as a later declaration may match it.

```

929         \def\reserved@a{#1}%
930         \ifx\pkgcls@targetlabel\reserved@a
931             \pkgcls@use@this@release{#3}{#2}%
932 (*tracerollback)
933         \else
934             \pkgcls@debug{Label doesn't match}%
935 (/tracerollback)
936         \fi
937     \fi
938 \fi
939 \fi
940 }

```

`\pkgcls@use@this@release` If a certain release has been selected (stored in the external file given in #1) we need to input it and afterwards stop reading the current file.

```

941 \def\pkgcls@use@this@release#1#2{%

```

Before that we record the selection made inside the transcript.

```

942     \pkgcls@show@selection{#1}{#2}%

```

We then set the `\pkgcls@targetdate` to zero so that any `\DeclareRelease` or `\DeclareCurrentRelease` in the file we now load are bypassed¹⁴ and then we finally load the correct release.

After loading that file we need to stop reading the current file so we issue `\endinput`. Note that the `\relax` before that is essential to ensure that the `\endinput` is only happening after the file has been fully processed, otherwise it would act after the first line of the `\@@input!`

```

943     \pkgcls@targetdate\z@

```

¹⁴The older release may also have such declarations inside if it was a simply copy of the `.sty` or `.cls` file current at that date. Removing these declarations would make the file load a tiny bit faster, but this way it works in any case.

```

944 \@@input #1\relax
945 \endinput
946 }

```

`\pkgcls@show@selection` This command records what selection was made. As that is needed in two places (and it is rather lengthy) it was placed in a separate command. The first argument is the name of the external file that is being loaded and is only needed for debugging. The second argument is the date that corresponds to this file and it is used as part of the message.

```

947 \def\pkgcls@show@selection#1#2{%
948 \*traceroollback)
949 \pkgcls@debug{Result: use #1}%
950 /traceroollback)
951 \GenericInfo
952 {\@spaces\@spaces\space}{Rollback for
953 \@cls@pkg\space'\@currname' requested ->
954 \ifnum\pkgcls@targetdate>\@ne
955     date
956     \ifnum\requestedLaTeXdate=\pkgcls@targetdate
957         \requestedpatchdate
958     \else
959         \expandafter\@gobble\pkgcls@arg
960     \fi.\MessageBreak

```

Instead of “best approximation” we could say that we have been able to exactly match the date (if it is exact), but that would mean extra tests without much gain, so not done.

```

961     Best approximation is
962 \else
963     version '\pkgcls@targetlabel'.\MessageBreak
964     This corresponds to
965 \fi
966 \ifx\@nil#2\@nil
967     a special release%
968 \else
969     the release introduced on #2%
970 \fi
971 \@gobble}%
972 }

```

`\pkgcls@rollbackdate@error` This is called if the requested rollback date is earlier than the earliest known release of a package or class.

A similar error is given if global rollback date and min-date on a specific package conflict with each other, but that case is happens only once so it is inlined.

```

973 \def\pkgcls@rollbackdate@error#1{%
974 \@latex@error{Suspicious rollback date given}%
975 {The \@cls@pkg\space'\@currname' claims that it
976     came into existence on #1 which\MessageBreak
977     is after your requested rollback date --- so
978     something is wrong here.\MessageBreak
979     Continue and we use the earliest known release.}}

```

\DeclareCurrentRelease This declares the date (and possible name) of the current version of a package or class.

```
980 \def\DeclareCurrentRelease#1#2{%
```

First we test if `\pkgcls@targetdate` is greater than zero, otherwise this code is bypassed (as there is no rollback request).

```
981 \ifnum\pkgcls@targetdate>\z@ % some sort of rollback request
982 (*tracrollback)
983 \pkgcls@debug{---DeclareCurrentRelease}%
984 \pkgcls@debug{ 1: #1}%
985 \pkgcls@debug{ 2: #2}%
986 /tracrollback}
```

If the value is greater than 1 we have to deal with a date request, so we parse #2 as a date and compare it with `\pkgcls@targetdate`.

```
987 \ifnum\pkgcls@targetdate>\@ne % a date request
988 \ifnum\@parse@version#2//00\@nil
989 >\pkgcls@targetdate
```

If it is greater that means the release date if this file is later than the requested rollback date. Again we have two cases: If there was a previous candidate release we use that one as the current release is too young, but if there wasn't we have to use this release nevertheless as there isn't any alternative.

However this case can only happen if there is a `\DeclareCurrentRelease` but no declared older releases (so basically the use of the declaration is a bit dubious).

```
990 \ifx\pkgcls@candidate\@empty
991 \pkgcls@rollbackdate@error{#2}%
992 \else
993 \pkgcls@use@this@release\pkgcls@candidate
994 \pkgcls@releasedate
995 \fi
```

Otherwise the current file is the right release, so we record that in the transcript and then carry on.

```
996 \else
997 \pkgcls@show@selection{current version}{#2}%
998 \fi
999 \else % a label request
```

Otherwise we have a rollback request to a named version so we check if that fits the current name and if not give an error as this was the last possible opportunity.

```
1000 \def\reserved@a{#1}%
1001 \ifx\pkgcls@targetlabel\reserved@a
1002 \pkgcls@show@selection{current version}{#2}%
1003 \else
1004 \@latex@error{Requested version '\pkgcls@targetlabel' for
1005 \cls@pkg\space'\@currname' is unknown}\@ehc
1006 \fi
1007 \fi
1008 \fi
1009 }
```

\IfTargetDateBefore This enables a simple form of conditional code inside a class or package file. If there is a date request and the request date is earlier than the first argument the code in the second argument is processed otherwise the code in the third argument

is processed. If there was no date request then we also execute the third argument, i.e., we will get the “latest” version of the file.

Most often the second argument (before-date-code) will be empty.

```

1010 \DeclareRobustCommand\IfTargetDateBefore[1]{%
1011   \ifnum\pkgcls@innerdate <%
1012     \expandafter\@parse@version\expandafter0#1//00\@nil
1013     \typeout{Exclude code introduced on #1}%
1014     \expandafter\@firstoftwo
1015   \else
1016     \typeout{Include code introduced on #1}%
1017     \expandafter\@secondoftwo
1018   \fi
1019 }

1020 </2ekernel | latexreleasefirst>

```

82 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```

1021 <*afterpreamble>
1022 \NeedsTeXFormat{LaTeX2e}
1023 \ProvidesPackage{pkgindoc}
1024   [1994/10/20 v1.1 Package Interface in Document (DPC)]
1025 \def\reserved@a#1\do\@classoptionslist#2\do\filec@ntents#3\relax{%
1026   \gdef\@preamblecmds{#1#3}}
1027 \expandafter\reserved@a\@preamblecmds\relax
1028 </afterpreamble>

```

File O

lthyphen.dtx

This file contains the code for loading hyphenation patterns into L^AT_EX. Most of this will end up in a file called `hyphen.ltx`. If you wish to customize your L^AT_EX 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^AT_EX 2_ε.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 \begin{document}
4 \DocInput{lthyphen.dtx}
5 \end{document}
6 \end{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_EX'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_EX run is terminated by invoking `\@@end` (which is the L^AT_EX 2_ε name for T_EX'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 \end{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 P

lualatex.dtx

83 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^AT_εX 2_ε kernel level plus as a loadable file which can be used with plain TeX and L^AT_εX.

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^AT_εX 2_ε kernel from 2015 onward these registers are part of the block in the extended area reserved by the kernel (prior to 2015 the L^AT_εX 2_ε kernel did not provide any functionality for the extended allocation area).

84 Core TeX functionality

The commands defined here are defined for possible inclusion in a future L^AT_εX format, however also extracted to the file `lualatex.tex` which may be used with older L^AT_εX formats, and with plain TeX.

<code>\newattribute</code>	<code>\newattribute{<attribute>}</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{<catcodetable>}</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{<function>}</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{<whatsit>}</code> Defines a custom <code>\whatsit</code> , indexed from 1.
<code>\newluabytecode</code>	<code>\newluabytecode{<bytecode>}</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.

85 Plain T_EX interface

The `luatex` interface may be used with plain T_EX using `\input{luatex}`. This inputs `luatex.tex` which inputs `etex.src` (or `etex.sty` if used with L^AT_EX) 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_EX, 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.

86 Lua functionality

86.1 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>-0x7FFFFFFF</code> ('unset'). The attribute allocation sequence is shared with the T _E 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 _E 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.
<code>new_luafunction</code>	<code>luatexbase.new_luafunction(⟨functionname⟩)</code> Returns an allocation number for a lua function for use with <code>\luafunction</code> , <code>\lateluafunction</code> , and <code>\luaodef</code> , indexed from 1. The optional <code>⟨functionname⟩</code> argument is just used for logging.

These functions all require access to a named T_EX count register to manage their allocations. The standard names are those defined above for access from T_EX, *e.g.* “e@alloc@attribute@count, but these can be adjusted by defining the variable `<type>_count_name` before loading `ltluatex.lua`, for example

```
local attribute_count_name = "attributetracker"
require("ltluatex")
```

would use a T_EX `\count` (`\countdef`’d token) called `attributetracker` in place of “e@alloc@attribute@count.

86.2 Lua access to T_EX register numbers

`registernumber` `luatexbase.registernumber(<name>)`

Sometimes (notably in the case of Lua attributes) it is necessary to access a register *by number* that has been allocated by T_EX. This package provides a function to look up the relevant number using LuaT_EX’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_AT_EX 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`).

86.3 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^AT_EX 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^AT_EX'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.

86.4 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 call-

back function with $\langle description \rangle$ from the $\langle callback \rangle$. The removed function and its description are returned as the results of this function.

<code>in_callback</code>	<code>luatexbase.in_callback($\langle callback \rangle$, $\langle description \rangle$)</code> Checks if the $\langle description \rangle$ matches one of the functions added to the list for the $\langle callback \rangle$, returning a boolean value.
<code>disable_callback</code>	<code>luatexbase.disable_callback($\langle callback \rangle$)</code> Sets the $\langle callback \rangle$ to <code>false</code> as described in the LuaTeX manual for the underlying <code>callback.register</code> built-in. Callbacks will only be set to <code>false</code> (and thus be skipped entirely) if there are no functions registered using the callback.
<code>callback_descriptions</code>	A list of the descriptions of functions registered to the specified callback is returned. <code>{}</code> is returned if there are no functions registered.
<code>create_callback</code>	<code>luatexbase.create_callback($\langle name \rangle$, $\langle metatype \rangle$, $\langle default \rangle$)</code> Defines a user defined callback. The last argument is a default function or <code>false</code> .
<code>call_callback</code>	<code>luatexbase.call_callback($\langle name \rangle$, ...)</code> Calls a user defined callback with the supplied arguments.

87 Implementation

```
1  $\langle *2ekernel | tex | latexrelease \rangle$ 
2  $\langle 2ekernel | latexrelease \rangle \ifx \directlua \@undefined \else$ 
```

87.1 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  $\langle latexrelease \rangle \backslash IncludeInRelease \{2015/10/01\}$ 
4  $\langle latexrelease \rangle \backslash newluafunction \{LuaTeX\} \%$ 
5  $\backslash ifnum \luatexversion < 60 \%$ 
6  $\backslash wlog \{*****\}$ 
7  $\backslash wlog \{* LuaTeX version too old for l\luatex support *\}$ 
8  $\backslash wlog \{*****\}$ 
9  $\backslash expandafter \endinput$ 
10  $\backslash fi$ 
```

87.2 Older L^AT_EX/Plain T_EX setup

```
11  $\langle *tex \rangle$ 
```

Older L^AT_EX formats don’t have the primitives with ‘native’ names: sort that out. If they already exist this will still be safe.

```
12  $\backslash directlua \{tex.enableprimitives("", tex.extraprimitives("luatex"))\}$ 
13  $\backslash ifx \e@alloc \@undefined$ 
14  $\backslash ifx \documentclass \@undefined$ 
15  $\backslash ifx \loccount \@undefined$ 
16  $\backslash input \{etex.src\} \%$ 
17  $\backslash fi$ 
18  $\backslash catcode '\@=11 \%$ 
```

In pre-2014 L^AT_EX, or plain T_EX, load `etex.{sty,src}`.

```

19 \outer\expandafter\def\csname newfam\endcsname
20 \csname fam\chardef\et@xmaxfam}
21 \else
22 \RequirePackage{etex}
23 \expandafter\def\csname newfam\endcsname
24 \csname fam\chardef\et@xmaxfam}
25 \expandafter\let\expandafter\new@mathgroup\csname newfam\endcsname
26 \fi

```

87.2.1 Fixes to etex.src/etex.sty

These could and probably should be made directly in an update to `etex.src` which already has some Lua_T_EX-specific code, but does not define the correct range for Lua_T_EX.

2015-07-13 higher range in luatex.

```
27 \edef \et@xmaxregs {\ifx\directlua\undefined 32768\else 65536\fi}
```

luatex/xetex also allow more math fam.

```

28 \edef \et@xmaxfam {\ifx\Umathchar\undefined\sixt@@n\else\ccclvi\fi}

29 \count 270=\et@xmaxregs % locally allocates \count registers
30 \count 271=\et@xmaxregs % ditto for \dimen registers
31 \count 272=\et@xmaxregs % ditto for \skip registers
32 \count 273=\et@xmaxregs % ditto for \muskip registers
33 \count 274=\et@xmaxregs % ditto for \box registers
34 \count 275=\et@xmaxregs % ditto for \toks registers
35 \count 276=\et@xmaxregs % ditto for \marks classes

```

and 256 or 16 fam. (Done above due to plain/L^AT_EX differences in l_tuatex.)

```
36 % \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}
```

End of proposed changes to `etex.src`

87.2.2 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.

```

37 \expandafter\let\csname newcount\expandafter\expandafter\endcsname
38 \csname globcount\endcsname
39 \expandafter\let\csname newdimen\expandafter\expandafter\endcsname
40 \csname globdimen\endcsname
41 \expandafter\let\csname newskip\expandafter\expandafter\endcsname
42 \csname globskip\endcsname
43 \expandafter\let\csname newbox\expandafter\expandafter\endcsname
44 \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.

```

45 \chardef\e@alloc@top=65535
46 \let\e@alloc\chardef\chardef

47 \def\e@alloc#1#2#3#4#5#6{%
48 \global\advance#3\@ne
49 \e@ch@ck{#3}{#4}{#5}#1%
50 \allocationnumber#3\relax
51 \global#2#6\allocationnumber

```

```

52 \wlog{\string#6=\string#1\the\allocationnumber}}%
53 \gdef\ech@ck#1#2#3#4{%
54 \ifnum#1<#2\else
55 \ifnum#1=#2\relax
56 #1\@cclvi
57 \ifx\count#4\advance#1 10 \fi
58 \fi
59 \ifnum#1<#3\relax
60 \else
61 \errmessage{No room for a new \string#4}%
62 \fi
63 \fi}%

```

Two simple L^AT_EX macros used in `lAtex.sty`.

```

64 \long\def\@gobble#1{}
65 \long\def\@firstofone#1{#1}

```

Fix up allocations not to clash with `etex.src`.

```

66 \expandafter\csname newcount\endcsname\@alloc@attribute@count
67 \expandafter\csname newcount\endcsname\@alloc@ccodetable@count
68 \expandafter\csname newcount\endcsname\@alloc@luafunction@count
69 \expandafter\csname newcount\endcsname\@alloc@whatsit@count
70 \expandafter\csname newcount\endcsname\@alloc@bytecode@count
71 \expandafter\csname newcount\endcsname\@alloc@luachunk@count

```

End of conditional setup for plain T_EX / old L^AT_EX.

```

72 \fi
73 \</tex>

```

87.3 Attributes

`\newattribute` As is generally the case for the LuaT_EX registers we start here from 1. Notably, some code assumes that `\attribute0` is never used so this is important in this case.

```

74 \ifx\@alloc@attribute@count\@undefined
75 \countdef\@alloc@attribute@count=258
76 \fi
77 \def\newattribute#1{%
78 \e@alloc@attribute\attributedef
79 \e@alloc@attribute@count\m@ne\@alloc@top#1%
80 }
81 \e@alloc@attribute@count=\z@

```

`\setattribute` Handy utilities.

```

\unsetattribute 82 \def\setattribute#1#2{#1=\numexpr#2\relax}
83 \def\unsetattribute#1{#1=-"7FFFFFFF\relax}

```

87.4 Category code tables

`\newcatcodetable` Category code tables are allocated with a limit half of that used by LuaT_EX 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.

```

84 \ifx\@alloc@ccodetable@count\@undefined
85 \countdef\@alloc@ccodetable@count=259
86 \fi
87 \def\newcatcodetable#1{%
88   \@alloc\catcodetable\chardef
89   \@alloc@ccodetable@count\m@ne{"8000}#1%
90   \initcatcodetable\allocationnumber
91 }
92 \@alloc@ccodetable@count=\z@

```

`\catcodetable@initex` Save a small set of standard tables. The Unicode data is read here in using a parser
`\catcodetable@string` simplified from that in load-unicode-data: only the nature of letters needs to
`\catcodetable@latex` be detected.
`\catcodetable@atletter`

```

93 \newcatcodetable\catcodetable@initex
94 \newcatcodetable\catcodetable@string
95 \begingroup
96   \def\setrangecatcode#1#2#3{%
97     \ifnum#1>#2 %
98       \expandafter\@gobble
99     \else
100       \expandafter\@firstofone
101     \fi
102     {%
103       \catcode#1=#3 %
104       \expandafter\setrangecatcode\expandafter
105       {\number\numexpr#1 + 1\relax}{#2}{#3}
106     }%
107   }
108   \@firstofone{%
109     \catcodetable\catcodetable@initex
110     \catcode0=12 %
111     \catcode13=12 %
112     \catcode37=12 %
113     \setrangecatcode{65}{90}{12}%
114     \setrangecatcode{97}{122}{12}%
115     \catcode92=12 %
116     \catcode127=12 %
117     \savecatcodetable\catcodetable@string
118   \endgroup
119   }%
120 \newcatcodetable\catcodetable@latex
121 \newcatcodetable\catcodetable@atletter
122 \begingroup
123   \def\parseunicodedataI#1;#2;#3;#4\relax{%
124     \parseunicodedataII#1;#3;#2 First>\relax
125   }%
126   \def\parseunicodedataII#1;#2;#3 First>#4\relax{%
127     \ifx\relax#4\relax
128       \expandafter\parseunicodedataIII
129     \else
130       \expandafter\parseunicodedataIV
131     \fi
132     {#1}#2\relax%

```

```

133 }%
134 \def\parseunicodedataIII#1#2#3\relax{%
135   \ifnum 0%
136     \if L#21\fi
137     \if M#21\fi
138     >0 %
139     \catcode"#1=11 %
140   \fi
141 }%
142 \def\parseunicodedataIV#1#2#3\relax{%
143   \read\unicoderead to \unicodedataline
144   \if L#2%
145     \count0="#1 %
146     \expandafter\parseunicodedataV\unicodedataline\relax
147   \fi
148 }%
149 \def\parseunicodedataV#1;#2\relax{%
150   \loop
151     \unless\ifnum\count0>"#1 %
152       \catcode\count0=11 %
153       \advance\count0 by 1 %
154   \repeat
155 }%
156 \def\storedpar{\par}%
157 \chardef\unicoderead=\numexpr\count16 + 1\relax
158 \openin\unicoderead=UnicodeData.txt %
159 \loop\unless\ifeof\unicoderead %
160   \read\unicoderead to \unicodedataline
161   \unless\ifx\unicodedataline\storedpar
162     \expandafter\parseunicodedataI\unicodedataline\relax
163   \fi
164 \repeat
165 \closein\unicoderead
166 \@firstofone{%
167   \catcode64=12 %
168   \savecatcodetable\catcodetable@latex
169   \catcode64=11 %
170   \savecatcodetable\catcodetable@atletter
171 }
172 \endgroup

```

87.5 Named Lua functions

`\newluafunction` Much the same story for allocating Lua_{TEX} functions except here they are just numbers so they are allocated in the same way as boxes. Lua indexes from 1 so once again slot 0 is skipped.

```

173 \ifx\e@alloc@luafunction@count\undefined
174   \countdef\e@alloc@luafunction@count=260
175 \fi
176 \def\newluafunction{%
177   \e@alloc@luafunction\e@alloc@chardef
178   \e@alloc@luafunction@count\m@ne\e@alloc@top
179 }

```

```
180 \e@alloc@luafunction@count=\z@
```

87.6 Custom whatsits

`\newwhatsit` These are only settable from Lua but for consistency are definable here.

```
181 \ifx\e@alloc@whatsit@count\@undefined
182 \countdef\e@alloc@whatsit@count=261
183 \fi
184 \def\newwhatsit#1{%
185   \e@alloc\whatsit\e@alloc@chardef
186   \e@alloc@whatsit@count\m@ne\e@alloc@top#1%
187 }
188 \e@alloc@whatsit@count=\z@
```

87.7 Lua bytecode registers

`\newluabytocode` These are only settable from Lua but for consistency are definable here.

```
189 \ifx\e@alloc@bytecode@count\@undefined
190 \countdef\e@alloc@bytecode@count=262
191 \fi
192 \def\newluabytocode#1{%
193   \e@alloc\luabytocode\e@alloc@chardef
194   \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
195 }
196 \e@alloc@bytecode@count=\z@
```

87.8 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.

```
197 \ifx\e@alloc@luachunk@count\@undefined
198 \countdef\e@alloc@luachunk@count=263
199 \fi
200 \def\newluachunkname#1{%
201   \e@alloc\luachunk\e@alloc@chardef
202   \e@alloc@luachunk@count\m@ne\e@alloc@top#1%
203   {\escapechar\m@ne
204    \directlua{lua.name[\the\allocationnumber]="\string#1"}}%
205 }
206 \e@alloc@luachunk@count=\z@
```

87.9 Lua loader

Load the Lua code at the start of every job. For the conversion of \TeX 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.

```
207 <2ekernel>\everyjob\expandafter{%
208 <2ekernel> \the\everyjob
209 \begingroup
210 \attributedef\attributezero=0 %
211 \chardef \charzero =0 %
```


Note name change required on older luatex, for hash table access.

```

212 \countdef \CountZero =0 %
213 \dimendef \dimenzero =0 %
214 \mathchardef \mathcharzero =0 %
215 \muskipdef \muskipzero =0 %
216 \skipdef \skipzero =0 %
217 \toksdef \tokszero =0 %
218 \directlua{require("ltnuatex")}
219 \endgroup
220 (2ekernel)}
221 \latexrelease\EndIncludeInRelease

222 \latexrelease\IncludeInRelease{0000/00/00}
223 \latexrelease {\newluafunction}{LuaTeX}%
224 \latexrelease\let\@alloc@attribute@count\@undefined
225 \latexrelease\let\newattribute\@undefined
226 \latexrelease\let\setattribute\@undefined
227 \latexrelease\let\unsetattribute\@undefined
228 \latexrelease\let\@alloc@ccodetable@count\@undefined
229 \latexrelease\let\newcatcodetable\@undefined
230 \latexrelease\let\catcodetable@initex\@undefined
231 \latexrelease\let\catcodetable@string\@undefined
232 \latexrelease\let\catcodetable@latex\@undefined
233 \latexrelease\let\catcodetable@atletter\@undefined
234 \latexrelease\let\@alloc@luafunction@count\@undefined
235 \latexrelease\let\newluafunction\@undefined
236 \latexrelease\let\@alloc@luafunction@count\@undefined
237 \latexrelease\let\newwhatsit\@undefined
238 \latexrelease\let\@alloc@whatsit@count\@undefined
239 \latexrelease\let\newluabytecode\@undefined
240 \latexrelease\let\@alloc@bytecode@count\@undefined
241 \latexrelease\let\newluachunkname\@undefined
242 \latexrelease\let\@alloc@luachunk@count\@undefined
243 \latexrelease\directlua{luatexbase.uninstall()}
244 \latexrelease\EndIncludeInRelease

In \everyjob, if luaotfload is available, load it and switch to TU.
245 \latexrelease\IncludeInRelease{2017/01/01}%
246 \latexrelease {\fontencoding}{TU in everyjob}%
247 \latexrelease\fontencoding{TU}\let\encodingdefault\fontencoding
248 \latexrelease\ifx\directlua\@undefined\else
249 (2ekernel)\everyjob\expandafter{%
250 (2ekernel) \the\everyjob
251 (*2ekernel, latexrelease)
252 \directlua{%
253 if xpcall(function ()%
254 require('luaotfload-main')%
255 end, texio.write_nl) then %
256 local _void = luaotfload.main ()%
257 else %
258 texio.write_nl('Error in luaotfload: reverting to OT1')%
259 tex.print('\string\def\string\encodingdefault{OT1}')%
260 end %
261 }%
262 \let\fontencoding\encodingdefault

```

```

263 \expandafter\let\csname ver@luaotfload.sty\endcsname\fmtversion
264 </2ekernel, latexrelease>
265 <latexrelease>\fi
266 <2ekernel> }
267 <latexrelease>\EndIncludeInRelease
268 <latexrelease>\IncludeInRelease{0000/00/00}%
269 <latexrelease>          {\fontencoding}{TU in everyjob}%
270 <latexrelease>\fontencoding{OT1}\let\encodingdefault\f@encoding
271 <latexrelease>\EndIncludeInRelease

272 <2ekernel | latexrelease>\fi
273 </2ekernel | tex | latexrelease>

```

87.10 Lua module preliminaries

```
274 <*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.

```

275 luatexbase      = luatexbase or { }
276 local luatexbase = luatexbase

```

Some Lua best practice: use local versions of functions where possible.

```

277 local string_gsub      = string.gsub
278 local tex_count        = tex.count
279 local tex_setattribute  = tex.setattribute
280 local tex_setcount     = tex.setcount
281 local texio_write_nl   = texio.write_nl

282 local luatexbase_warning
283 local luatexbase_error

```

87.11 Lua module utilities

87.11.1 Module tracking

modules To allow tracking of module usage, a structure is provided to store information and to return it.

```
284 local modules = modules or { }
```

provides_module Local function to write to the log.

```

285 local function luatexbase_log(text)
286   texio_write_nl("log", text)
287 end

```

Modelled on `\ProvidesPackage`, we store much the same information but with a little more structure.

```

288 local function provides_module(info)
289   if not (info and info.name) then
290     luatexbase_error("Missing module name for provides_module")
291   end
292   local function spaced(text)
293     return text and (" " .. text) or ""

```

```

294 end
295 luatexbase_log(
296   "Lua module: " .. info.name
297   .. spaced(info.date)
298   .. spaced(info.version)
299   .. spaced(info.description)
300 )
301 modules[info.name] = info
302 end
303 luatexbase.provides_module = provides_module

```

87.11.2 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.

```

304 local function msg_format(mod, msg_type, text)
305   local leader = ""
306   local cont
307   local first_head
308   if mod == "LaTeX" then
309     cont = string_gsub(leader, ".", " ")
310     first_head = leader .. "LaTeX: "
311   else
312     first_head = leader .. "Module " .. msg_type
313     cont = "(" .. mod .. ")"
314     .. string_gsub(first_head, ".", " ")
315     first_head = leader .. "Module " .. mod .. " " .. msg_type .. ":"
316   end
317   if msg_type == "Error" then
318     first_head = "\n" .. first_head
319   end
320   if string.sub(text,-1) ~= "\n" then
321     text = text .. " "
322   end
323   return first_head .. " "
324   .. string_gsub(
325     text
326     .. "on input line "
327     .. tex.inputlineno, "\n", "\n" .. cont .. " "
328   )
329   .. "\n"
330 end

```

```

module_info Write messages.
module_warning 331 local function module_info(mod, text)
module_error 332   texio_write_nl("log", msg_format(mod, "Info", text))
333 end
334 luatexbase.module_info = module_info

```

```

335 local function module_warning(mod, text)
336   texio_write_nl("term and log",msg_format(mod, "Warning", text))
337 end
338 luatexbase.module_warning = module_warning
339 local function module_error(mod, text)
340   error(msg_format(mod, "Error", text))
341 end
342 luatexbase.module_error = module_error

```

Dedicated versions for the rest of the code here.

```

343 function luatexbase_warning(text)
344   module_warning("luatexbase", text)
345 end
346 function luatexbase_error(text)
347   module_error("luatexbase", text)
348 end

```

87.12 Accessing register numbers from Lua

Collect up the data from the T_EX level into a Lua table: from version 0.80, LuaT_EX makes that easy.

```

349 local luaregisterbasetable = { }
350 local registermap = {
351   attributezero = "assign_attr"   ,
352   charzero      = "char_given"    ,
353   CountZero     = "assign_int"     ,
354   dimenzero     = "assign_dimen"   ,
355   mathcharzero  = "math_given"     ,
356   muskipzero    = "assign_mu_skip" ,
357   skipzero      = "assign_skip"    ,
358   tokszero      = "assign_toks"    ,
359 }
360 local createtoken
361 if tex.luatexversion > 81 then
362   createtoken = token.create
363 elseif tex.luatexversion > 79 then
364   createtoken = newtoken.create
365 end
366 local hashtokens = tex.hashtokens()
367 local luatexversion = tex.luatexversion
368 for i,j in pairs (registermap) do
369   if luatexversion < 80 then
370     luaregisterbasetable[hashtokens[i][1]] =
371       hashtokens[i][2]
372   else
373     luaregisterbasetable[j] = createtoken(i).mode
374   end
375 end

```

registernumber Working out the correct return value can be done in two ways. For older LuaT_EX releases it has to be extracted from the `hashtokens`. On the other hand, newer LuaT_EX's have `newtoken`, and whilst `.mode` isn't currently documented, Hans Hagen pointed to this approach so we should be OK.

```

376 local registernumber
377 if luatexversion < 80 then
378   function registernumber(name)
379     local nt = hashtokens[name]
380     if(nt and luaregisterbasetable[nt[1]]) then
381       return nt[2] - luaregisterbasetable[nt[1]]
382     else
383       return false
384     end
385   end
386 else
387   function registernumber(name)
388     local nt = createtoken(name)
389     if(luaregisterbasetable[nt.cmdname]) then
390       return nt.mode - luaregisterbasetable[nt.cmdname]
391     else
392       return false
393     end
394   end
395 end
396 luatexbase.registernumber = registernumber

```

87.13 Attribute allocation

new_attribute As attributes are used for Lua manipulations its useful to be able to assign from this end.

```

397 local attributes=setmetatable(
398 {},
399 {
400   __index = function(t,key)
401     return registernumber(key) or nil
402   end}
403 )
404 luatexbase.attributes = attributes
405 local attribute_count_name =
406   attribute_count_name or "e@alloc@attribute@count"
407 local function new_attribute(name)
408   tex_setcount("global", attribute_count_name,
409     tex_count[attribute_count_name] + 1)
410   if tex_count[attribute_count_name] > 65534 then
411     luatexbase_error("No room for a new \\attribute")
412   end
413   attributes[name]= tex_count[attribute_count_name]
414   luatexbase_log("Lua-only attribute " .. name .. " = " ..
415     tex_count[attribute_count_name])
416   return tex_count[attribute_count_name]
417 end
418 luatexbase.new_attribute = new_attribute

```

87.14 Custom whatsit allocation

new_whatsit Much the same as for attribute allocation in Lua.

```

419 local whatsit_count_name = whatsit_count_name or "e@alloc@whatsit@count"

```

```

420 local function new_whatsit(name)
421   tex_setcount("global", whatsit_count_name,
422               tex_count[whatsit_count_name] + 1)
423   if tex_count[whatsit_count_name] > 65534 then
424     luatexbase_error("No room for a new custom whatsit")
425   end
426   luatexbase_log("Custom whatsit " .. (name or "") .. " = " ..
427                 tex_count[whatsit_count_name])
428   return tex_count[whatsit_count_name]
429 end
430 luatexbase.new_whatsit = new_whatsit

```

87.15 Bytecode register allocation

new_bytecode Much the same as for attribute allocation in Lua. The optional *<name>* argument is used in the log if given.

```

431 local bytecode_count_name =
432     bytecode_count_name or "e@alloc@bytecode@count"
433 local function new_bytecode(name)
434   tex_setcount("global", bytecode_count_name,
435               tex_count[bytecode_count_name] + 1)
436   if tex_count[bytecode_count_name] > 65534 then
437     luatexbase_error("No room for a new bytecode register")
438   end
439   luatexbase_log("Lua bytecode " .. (name or "") .. " = " ..
440                 tex_count[bytecode_count_name])
441   return tex_count[bytecode_count_name]
442 end
443 luatexbase.new_bytecode = new_bytecode

```

87.16 Lua chunk name allocation

new_chunkname As for bytecode registers but also store the name in the `lua.name` table.

```

444 local chunkname_count_name =
445     chunkname_count_name or "e@alloc@luachunk@count"
446 local function new_chunkname(name)
447   tex_setcount("global", chunkname_count_name,
448               tex_count[chunkname_count_name] + 1)
449   local chunkname_count = tex_count[chunkname_count_name]
450   chunkname_count = chunkname_count + 1
451   if chunkname_count > 65534 then
452     luatexbase_error("No room for a new chunkname")
453   end
454   lua.name[chunkname_count]=name
455   luatexbase_log("Lua chunkname " .. (name or "") .. " = " ..
456                 chunkname_count .. "\n")
457   return chunkname_count
458 end
459 luatexbase.new_chunkname = new_chunkname

```

87.17 Lua function allocation

`new_luafunction` Much the same as for attribute allocation in Lua. The optional $\langle name \rangle$ argument is used in the log if given.

```
460 local luafunction_count_name =
461     luafunction_count_name or "e@alloc@luafunction@count"
462 local function new_luafunction(name)
463     tex_setcount("global", luafunction_count_name,
464         tex_count[luafunction_count_name] + 1)
465     if tex_count[luafunction_count_name] > 65534 then
466         luatexbase_error("No room for a new luafunction register")
467     end
468     luatexbase_log("Lua function " .. (name or "") .. " = " ..
469         tex_count[luafunction_count_name])
470     return tex_count[luafunction_count_name]
471 end
472 luatexbase.new_luafunction = new_luafunction
```

87.18 Lua callback management

The native mechanism for callbacks in LuaTeX allows only one per function. That is extremely restrictive and so a mechanism is needed to add and remove callbacks from the appropriate hooks.

87.18.1 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.

```
473 local callbacklist = callbacklist or { }
```

Numerical codes for callback types, and name-to-value association (the table keys are strings, the values are numbers).

```
474 local list, data, exclusive, simple = 1, 2, 3, 4
475 local types = {
476     list      = list,
477     data      = data,
478     exclusive = exclusive,
479     simple    = simple,
480 }
```

Now, list all predefined callbacks with their current type, based on the LuaTeX manual version 1.01. 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.)

481 local callbacktypes = callbacktypes or {

Section 8.2: file discovery callbacks.

```
482 find_read_file      = exclusive,
483 find_write_file     = exclusive,
484 find_font_file      = data,
485 find_output_file    = data,
486 find_format_file    = data,
487 find_vf_file        = data,
488 find_map_file       = data,
489 find_enc_file       = data,
490 find_pk_file        = data,
491 find_data_file      = data,
492 find_opentype_file  = data,
493 find_truetype_file  = data,
494 find_type1_file     = data,
495 find_image_file     = data,
```

```
496 open_read_file      = exclusive,
497 read_font_file      = exclusive,
498 read_vf_file        = exclusive,
499 read_map_file       = exclusive,
500 read_enc_file       = exclusive,
501 read_pk_file        = exclusive,
502 read_data_file      = exclusive,
503 read_truetype_file  = exclusive,
504 read_type1_file     = exclusive,
505 read_opentype_file  = exclusive,
```

Not currently used by luatex but included for completeness. may be used by a font handler.

```
506 find_cidmap_file   = data,
507 read_cidmap_file   = exclusive,
```

Section 8.3: data processing callbacks.

```
508 process_input_buffer = data,
509 process_output_buffer = data,
510 process_jobname      = data,
```

Section 8.4: node list processing callbacks.

```
511 contribute_filter    = simple,
512 buildpage_filter     = simple,
513 build_page_insert    = exclusive,
514 pre_linebreak_filter = list,
515 linebreak_filter     = exclusive,
516 append_to_vlist_filter = exclusive,
517 post_linebreak_filter = list,
518 hpack_filter         = list,
519 vpack_filter         = list,
520 hpack_quality        = list,
521 vpack_quality        = list,
522 pre_output_filter    = list,
523 process_rule         = exclusive,
```



```

524 hyphenate          = simple,
525 ligaturing         = simple,
526 kerning            = simple,
527 insert_local_par   = simple,
528 mlist_to_hlist     = exclusive,
529 new_graf           = simple,

```

Section 8.5: information reporting callbacks.

```

530 pre_dump           = simple,
531 start_run           = simple,
532 stop_run            = simple,
533 start_page_number  = simple,
534 stop_page_number   = simple,
535 show_error_hook    = simple,
536 show_warning_message = simple,
537 show_error_message = simple,
538 show_lua_error_hook = simple,
539 start_file          = simple,
540 stop_file           = simple,
541 call_edit           = simple,
542 finish_synctex      = simple,
543 wrapup_run          = simple,

```

Section 8.6: PDF-related callbacks.

```

544 finish_pdffile      = data,
545 finish_pdfpage      = data,
546 page_objnum_provider = data,
547 process_pdf_image_content = data,

```

Section 8.7: font-related callbacks.

```

548 define_font         = exclusive,
549 glyph_not_found     = exclusive,
550 glyph_stream_provider = exclusive,
551 make_extensible     = exclusive,
552 font_descriptor_objnum_provider = exclusive,

```

```

553 }

```

```

554 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.

```

555 local callback_register = callback_register or callback.register
556 function callback.register()
557   luatexbase_error("Attempt to use callback.register() directly\n")
558 end

```

87.18.2 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, the 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.

The way the functions are combined together depends on the type of the callback. There are currently 4 types of callback, depending on the calling convention of the functions the callback can hold:

simple is for functions that don't return anything: they are called in order, all with the same argument;

data is for functions receiving a piece of data of any type except node list head (and possibly other arguments) and returning it (possibly modified): the functions are called in order, and each is passed the return value of the previous (and the other arguments untouched, if any). The return value is that of the last function;

list is a specialized variant of *data* for functions filtering node lists. Such functions may return either the head of a modified node list, or the boolean values **true** or **false**. The functions are chained the same way as for *data* except that for the following. If one function returns **false**, then **false** is immediately returned and the following functions are *not* called. If one function returns **true**, then the same head is passed to the next function. If all functions return **true**, then **true** is returned, otherwise the return value of the last function not returning **true** is used.

exclusive is for functions with more complex signatures; functions in this type of callback are *not* combined: An error is raised if a second callback is registered..

Handler for **data** callbacks.

```
559 local function data_handler(name)
560   return function(data, ...)
561     for _,i in ipairs(callbacklist[name]) do
562       data = i.func(data,...)
563     end
564     return data
565   end
566 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.

```
567 local function exclusive_handler(name)
568   return function(...)
569     return callbacklist[name][1].func(...)
570   end
571 end
```

Handler for **list** callbacks.

```
572 local function list_handler(name)
573   return function(head, ...)
574     local ret
575     local alltrue = true
576     for _,i in ipairs(callbacklist[name]) do
577       ret = i.func(head, ...)
578       if ret == false then
579         luatexbase_warning(
580           "Function '" .. i.description .. "' returned false\n"
```

```

581         .. "in callback '" .. name ..'"
582     )
583     break
584 end
585 if ret ~= true then
586     alltrue = false
587     head = ret
588 end
589 end
590 return alltrue and true or head
591 end
592 end

```

Handler for simple callbacks.

```

593 local function simple_handler(name)
594     return function(...)
595         for _,i in ipairs(callbacklist[name]) do
596             i.func(...)
597         end
598     end
599 end

```

Keep a handlers table for indexed access.

```

600 local handlers = {
601     [data]      = data_handler,
602     [exclusive] = exclusive_handler,
603     [list]      = list_handler,
604     [simple]     = simple_handler,
605 }

```

87.18.3 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, it may be declared as `false`. First we need a list of user callbacks.

```

606 local user_callbacks_defaults = { }

```

`create_callback` The allocator itself.

```

607 local function create_callback(name, ctype, default)
608     if not name or name == ""
609     or not ctype or ctype == ""
610     then
611         luatexbase_error("Unable to create callback:\n" ..
612             "valid callback name and type required")
613     end
614     if callbacktypes[name] then
615         luatexbase_error("Unable to create callback '" .. name ..
616             "':\ncallback is already defined")
617     end
618     if default ~= false and type (default) ~= "function" then
619         luatexbase_error("Unable to create callback '" .. name ..
620             "':\ndefault is not a function")
621     end
622     user_callbacks_defaults[name] = default

```

```

623  callbacktypes[name] = types[ctype]
624 end
625 luatexbase.create_callback = create_callback

call_callback  Call a user defined callback. First check arguments.
626 local function call_callback(name,...)
627   if not name or name == "" then
628     luatexbase_error("Unable to create callback:\n" ..
629       "valid callback name required")
630   end
631   if user_callbacks_defaults[name] == nil then
632     luatexbase_error("Unable to call callback '" .. name
633       .. "':\nunknown or empty")
634   end
635   local l = callbacklist[name]
636   local f
637   if not l then
638     f = user_callbacks_defaults[name]
639     if l == false then
640       return nil
641     end
642   else
643     f = handlers[callbacktypes[name]](name)
644   end
645   return f(...)
646 end
647 luatexbase.call_callback=call_callback

add_to_callback  Add a function to a callback. First check arguments.
648 local function add_to_callback(name, func, description)
649   if not name or name == "" then
650     luatexbase_error("Unable to register callback:\n" ..
651       "valid callback name required")
652   end
653   if not callbacktypes[name] or
654     type(func) ~= "function" or
655     not description or
656     description == "" then
657     luatexbase_error(
658       "Unable to register callback.\n\n"
659       .. "Correct usage:\n"
660       .. "add_to_callback(<callback>, <function>, <description>)"
661     )
662   end

  Then test if this callback is already in use. If not, initialise its list and register the
  proper handler.
663   local l = callbacklist[name]
664   if l == nil then
665     l = { }
666     callbacklist[name] = l

  If it is not a user defined callback use the primitive callback register.
667   if user_callbacks_defaults[name] == nil then
668     callback_register(name, handlers[callbacktypes[name]](name))

```

```

669     end
670 end

```

Actually register the function and give an error if more than one **exclusive** one is registered.

```

671 local f = {
672     func      = func,
673     description = description,
674 }
675 local priority = #l + 1
676 if callbacktypes[name] == exclusive then
677     if #l == 1 then
678         luatexbase_error(
679             "Cannot add second callback to exclusive function\n'" ..
680             name .. "'"")
681     end
682 end
683 table.insert(l, priority, f)

```

Keep user informed.

```

684 luatexbase_log(
685     "Inserting '" .. description .. "' at position "
686     .. priority .. " in '" .. name .. "'"")
687 )
688 end
689 luatexbase.add_to_callback = add_to_callback

```

remove_from_callback Remove a function from a callback. First check arguments.

```

690 local function remove_from_callback(name, description)
691     if not name or name == "" then
692         luatexbase_error("Unable to remove function from callback:\n" ..
693             "valid callback name required")
694     end
695     if not callbacktypes[name] or
696         not description or
697         description == "" then
698         luatexbase_error(
699             "Unable to remove function from callback.\n\n"
700             .. "Correct usage:\n"
701             .. "remove_from_callback(<callback>, <description>)"
702         )
703     end
704     local l = callbacklist[name]
705     if not l then
706         luatexbase_error(
707             "No callback list for '" .. name .. "'\n")
708     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.

```

709     local index = false
710     for i,j in ipairs(l) do
711         if j.description == description then
712             index = i
713             break

```

```

714     end
715 end
716 if not index then
717     luatexbase_error(
718         "No callback '" .. description .. "' registered for '" ..
719         name .. "'\n")
720 end
721 local cb = l[index]
722 table.remove(l, index)
723 luatexbase_log(
724     "Removing '" .. description .. "' from '" .. name .. "'."
725 )
726 if #l == 0 then
727     callbacklist[name] = nil
728     callback_register(name, nil)
729 end
730 return cb.func,cb.description
731 end
732 luatexbase.remove_from_callback = remove_from_callback

```

in_callback Look for a function description in a callback.

```

733 local function in_callback(name, description)
734     if not name
735     or name == ""
736     or not callbacklist[name]
737     or not callbacktypes[name]
738     or not description then
739         return false
740     end
741     for _, i in pairs(callbacklist[name]) do
742         if i.description == description then
743             return true
744         end
745     end
746     return false
747 end
748 luatexbase.in_callback = in_callback

```

disable_callback As we subvert the engine interface we need to provide a way to access this functionality.

```

749 local function disable_callback(name)
750     if(callbacklist[name] == nil) then
751         callback_register(name, false)
752     else
753         luatexbase_error("Callback list for " .. name .. " not empty")
754     end
755 end
756 luatexbase.disable_callback = disable_callback

```

callback_descriptions List the descriptions of functions registered for the given callback.

```

757 local function callback_descriptions (name)
758     local d = {}
759     if not name
760     or name == ""

```

```

761     or not callbacklist[name]
762     or not callbacktypes[name]
763     then
764         return d
765     else
766         for k, i in pairs(callbacklist[name]) do
767             d[k]= i.description
768         end
769     end
770     return d
771 end
772 luatexbase.callback_descriptions =callback_descriptions

```

`uninstall` Unlike at the T_EX 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!

```

773 local function uninstall()
774     module_info(
775         "luatexbase",
776         "Uninstalling kernel luatexbase code"
777     )
778     callback.register = callback_register
779     luatexbase = nil
780 end
781 luatexbase.uninstall = uninstall
782  $\langle$ /lua $\rangle$ 

```

Reset the catcode of @.

```

783  $\langle$ tex $\rangle$ \catcode'\@=\etatcatcode\relax

```

File Q

ltfinal.dtx

88 Final settings

This section contains the final settings for L^AT_EX. It initialises some debugging and typesetting parameters, sets the default `\catcodes` and `uc/lc` codes, and inputs the hyphenation file.

88.1 Debugging

By default, L^AT_EX shows statistics:

```
1 \*2kernel)
2 \tracingstats1
```

88.2 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 \*2kernel)
7 \*2kernel | latexrelease)
8 \*2kernel) \IncludeInRelease{2015/01/01}%
9 \*2kernel) \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 \*2kernel | latexrelease)
15 \*2kernel) \EndIncludeInRelease
16 \*2kernel) \IncludeInRelease{0000/00/00}%
17 \*2kernel) \newmarks{Extended Allocation}%
18 \*2kernel) \let\newmarks\@undefined
19 \*2kernel) \EndIncludeInRelease
20 \*2kernel)

\newXeTeXintercharclass Allocate \XeTeXintercharclass types if xetex is active. previously defined in
\XeTeXintercharclass xetex.ini.
\XeTeXintercharclass
\XeTeXintercharclass@top
21 \*2kernel)
22 \*2kernel | latexrelease)
23 \*2kernel) \IncludeInRelease{2015/01/01}%
24 \*2kernel) \newXeTeXintercharclass{Extended Allocation}%

Classes allocated 1 to 4094 (or 254 on older xetex) (In earlier XeLaTeX versions
1, 2 and 3 were pre-set for CJK).
25 \ifx\XeTeXcharclass\@undefined
26 \else
```



```

27 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
28   \chardef\Xe@alloc@intercharclass@top=4095
29 \else
30   \chardef\Xe@alloc@intercharclass@top=255
31 \fi

32 \def\newXeTeXintercharclass{%
33   \Xe@alloc\XeTeXcharclass
34   \chardef\Xe@alloc@intercharclass\m@ne\Xe@alloc@intercharclass@top}
35 \fi

36 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
37 \else
38 \fi
39 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
40 \else
41 \fi
42 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
43 \else
44 \fi
45 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
46 \else
47 \fi
48 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
49 \else
50 \fi
51 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
52 \else
53 \fi
54 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
55 \else
56 \fi
57 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
58 \else
59 \fi
60 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
61 \else
62 \fi
63 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
64 \else
65 \fi
66 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
67 \else
68 \fi
69 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
70 \else
71 \fi
72 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
73 \else
74 \fi
75 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
76 \else
77 \fi
78 \ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
79 \else

```

```

80 <latexrelease>\EndIncludeInRelease
81 <*2ekernel>

```

The default values of the picture and \fbox parameters:

```

82 \unitlength = 1pt
83 \fboxsep = 3pt
84 \fboxrule = .4pt

```

The saved value of T_EX's \maxdepth:

```

85 \@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}

```

86 \vsize = 1000pt
87 \@colroom = \vsize
88 \@colht = \vsize

```

Initialise \textheight \textwidth and page style, to avoid internal errors if they
 are not set by the class.

```

89 \textheight=.5\maxdimen
90 \textwidth=\textheight
91 \ps@empty

```

88.3 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 common loader system provides access to data from the Unicode Consortium covering not only \lccode but also other related data. The \lccode part of that at least needs to be loaded before hyphenation is tackled: XeT_EX follows the standard T_EX route of building patterns into the format. LuaT_EX 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.

```

92 \ifnum 0%
93   \ifx\Umathcode\@undefined\else 1\fi
94   \ifx\XeTeXmathcode\@undefined\else 1\fi
95   >\z@
96   \message{ Unicode character data,}
97   \input{load-unicode-data}
98 </2ekernel>
99 <latexrelease>\IncludeInRelease{2016/02/01}%
100 <latexrelease> {\XeTeXintercharclasses}{XeTeX character classes}%
101 <latexrelease> \ifx\XeTeXinterchartoks\undefined
102 <latexrelease> \else
103 <latexrelease>   \begingroup
104 <latexrelease>     \chardef\XeTeXcharclassID = 0 %
105 <latexrelease>     \chardef\XeTeXcharclassOP = 0 %
106 <latexrelease>     \chardef\XeTeXcharclassCL = 0 %
107 <latexrelease>     \chardef\XeTeXcharclassEX = 0 %
108 <latexrelease>     \chardef\XeTeXcharclassIS = 0 %

```

```

109 \latexrelease \chardef\XeTeXcharclassNS = 0 %
110 \latexrelease \chardef\XeTeXcharclassCM = 0 %
111 \latexrelease \input{load-unicode-xetex-classes}
112 \latexrelease \endgroup
113 \latexrelease \global\let\xtxHanGlue\undefined
114 \latexrelease \global\let\xtxHanSpace\undefined
115 \latexrelease \global\XeTeXinterchartoks 0 1 = {}
116 \latexrelease \global\XeTeXinterchartoks 0 2 = {}
117 \latexrelease \global\XeTeXinterchartoks 0 3 = {}
118 \latexrelease \global\XeTeXinterchartoks 1 0 = {}
119 \latexrelease \global\XeTeXinterchartoks 2 0 = {}
120 \latexrelease \global\XeTeXinterchartoks 3 0 = {}
121 \latexrelease \global\XeTeXinterchartoks 1 1 = {}
122 \latexrelease \global\XeTeXinterchartoks 1 2 = {}
123 \latexrelease \global\XeTeXinterchartoks 1 3 = {}
124 \latexrelease \global\XeTeXinterchartoks 2 1 = {}
125 \latexrelease \global\XeTeXinterchartoks 2 2 = {}
126 \latexrelease \global\XeTeXinterchartoks 2 3 = {}
127 \latexrelease \global\XeTeXinterchartoks 3 1 = {}
128 \latexrelease \global\XeTeXinterchartoks 3 2 = {}
129 \latexrelease \global\XeTeXinterchartoks 3 3 = {}
130 \latexrelease \fi
131 \latexrelease \EndIncludeInRelease
132 \latexrelease \IncludeInRelease{0000/00/00}%
133 \latexrelease {\XeTeXintercharclasses}{XeTeX character classes}%
134 \latexrelease \ifx\XeTeXinterchartoks\undefined
135 \latexrelease \else
136 \latexrelease \input{load-unicode-xetex-classes}
137 \latexrelease \gdef\xtxHanGlue{\hskip0pt plus 0.1em\relax}
138 \latexrelease \gdef\xtxHanSpace{\hskip0.2em plus 0.2em minus 0.1em\relax}
139 \latexrelease \global\XeTeXinterchartoks 0 1 = {\xtxHanSpace}
140 \latexrelease \global\XeTeXinterchartoks 0 2 = {\xtxHanSpace}
141 \latexrelease \global\XeTeXinterchartoks 0 3 = {\nobreak\xtxHanSpace}
142 \latexrelease \global\XeTeXinterchartoks 1 0 = {\xtxHanSpace}
143 \latexrelease \global\XeTeXinterchartoks 2 0 = {\nobreak\xtxHanSpace}
144 \latexrelease \global\XeTeXinterchartoks 3 0 = {\xtxHanSpace}
145 \latexrelease \global\XeTeXinterchartoks 1 1 = {\xtxHanGlue}
146 \latexrelease \global\XeTeXinterchartoks 1 2 = {\xtxHanGlue}
147 \latexrelease \global\XeTeXinterchartoks 1 3 = {\nobreak\xtxHanGlue}
148 \latexrelease \global\XeTeXinterchartoks 2 1 = {\nobreak\xtxHanGlue}
149 \latexrelease \global\XeTeXinterchartoks 2 2 = {\nobreak\xtxHanGlue}
150 \latexrelease \global\XeTeXinterchartoks 2 3 = {\xtxHanGlue}
151 \latexrelease \global\XeTeXinterchartoks 3 1 = {\xtxHanGlue}
152 \latexrelease \global\XeTeXinterchartoks 3 2 = {\xtxHanGlue}
153 \latexrelease \global\XeTeXinterchartoks 3 3 = {\nobreak\xtxHanGlue}
154 \latexrelease \fi
155 \latexrelease \EndIncludeInRelease
156 \kernel

```

There is one over-ride that makes sense here (see below for the same for 8-bit engines): setting the lccode for - to itself.

```
157 \lccode'\- ='\- % default hyphen char
```

The alternative is that a “traditional” engine is in use.

```
158 \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.

```

159 \def\reserved@a#1#2{%
160   \@tempcnta#1\relax
161   \@tempcntb#2\relax
162   \reserved@b
163 }
164 \def\reserved@b{%
165   \ifnum\@tempcnta>\@tempcntb\else
166     \reserved@c\@tempcnta
167     \advance\@tempcnta\@ne
168     \expandafter\reserved@b
169   \fi
170 }

```

Depending on the T_EX version, we might not be allowed to do this for non-ASCII characters.

```

171 \def\reserved@c#1{%
172   \count@=#1\advance\count@ by -"20
173   \uccode#1=\count@
174   \lccode#1=#1
175 }
176 \reserved@a{'\a}{'\z}
177 \reserved@a{"A0}{"BC}
178 \reserved@a{"E0}{"FF}

```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```

179 \def\reserved@c#1{%
180   \count@=#1\advance\count@ by "20
181   \uccode#1=#1
182   \lccode#1=\count@
183   \sfcode#1=999
184 }
185 \reserved@a{'\A}{'\Z}
186 \reserved@a{"80}{"9C}
187 \reserved@a{"C0}{"DF}

```

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.

```

188 \uccode'\^^Y='^I      % dotless i
189 \lccode'\^^Y='^Y      % dotless i
190 \uccode'\^^Z='^J      % dotless j, ae in OT1
191 \lccode'\^^Z='^Z      % dotless j, ae in OT1
192 \lccode'\^^9d='^i     % dotted I
193 \uccode'\^^9d='^9d    % dotted I
194 \lccode'\^^9e='^9e    % d-bar
195 \uccode'\^^9e='^d0    % d-bar

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

196 \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).

```
197 \lccode'\- ='\- % default hyphen char
198 \lccode 127=127 % alternate hyphen char
199 \lccode 23 =23 % textcompwordmark in T1
```

End of the conditional to select either Unicode or T1 encoding defaults.

```
200 \fi
```

This is as good a place as any to active a few XeTeX-specific settings

```
201 \ifx\XeTeXuseglyphmetrics\@undefined
202 \else
203   \XeTeXuseglyphmetrics=1 %
204   \XeTeXdashbreakstate=1 %
205 \fi
```

88.4 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.

```
206 \InputIfFileExists{hyphen.cfg}
207   {\typeout{=====^~J%
208             Local configuration file hyphen.cfg used^~J%
209             =====}%
210   \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
211   }
212   {\input{hyphen.ltx}}
213 \let\@addtofilelist\@gobble
```

`\l@nohyphenation`

```
214 \ifx\l@nohyphenation \@undefined
215   \newlanguage\l@nohyphenation
216 \fi
```

`\document@default@language` Default document language. -1 acts as language 0, but used as a flag in `\document` to see if it has been set in the preamble.

```
217 </2ekernel>
218 <*2ekernel | latexrelease>
219 <latexrelease>\IncludeInRelease{2017/04/15}%
220 <latexrelease>          {\document@default@language}{Save language for hyphenation}%
221 \let\document@default@language\m@ne
222 </2ekernel | latexrelease>
223 <latexrelease>\EndIncludeInRelease
224 <latexrelease>\IncludeInRelease{0000/00/00}%
225 <latexrelease>          {\document@default@language}{Save language for hyphenation}%
226 <latexrelease>\let\document@default@language\@undefined
227 <latexrelease>\EndIncludeInRelease
228 <*2ekernel>
```

88.5 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.

```
229 \ifdim \font@submax >\z@
230   \@font@warning{Size substitutions with differences\MessageBreak
231                 up to \font@submax\space have occurred.\MessageBreak
232                 \MessageBreak
233                 Please check the transcript file
234                 carefully\MessageBreak
235                 and redo the format generation if necessary!
236                 \@gobbletwo}%
237 \errhelp{Only stopped, to give you time to
238         read the above message.}
239 \errmessage{}
```

We reset the macro. Otherwise every user will get a warning on every job.

```
240 \def\font@submax{0pt}
241 \fi
```

88.6 Input encoding

Starting with the 2018 L^AT_EX release default the inputencoding to UTF-8. Unless the format is being used with luatex, xetex, enc_{te}x or ml_{te}x.

This is done in a way largely compatible with older releases: `utf8.def` is input just as if

```
\usepackage[utf8]{inputenc}
```

had been used, however rather than input the whole package a minimal core part just enough to support loading the UTF-8 encoding files is defined here.

If a document re-specifies UTF-8 this is silently ignored.

```
242 \ifx\kernel
243 (*2kernel | latexrelease)
```

Check that a classic 8-bit tex engine is being used (LaTeX or PDFLaTeX).

```
244 \ifx\kernel\includeinrelease{2018/04/01}%
245 \ifx\kernel\includeinrelease{UTFviii@invalid}{UTF-8 default}%
```

Skip this section in Unicode TeX, or if MLTeX and EncTeX are enabled.

```
246 \ifnum0%
247   \ifx\Umathchar\@undefined\else 1\fi
248   \ifx\mubyte\@undefined\else 1\fi
249   \ifx\charsubdef\@undefined\else 1\fi
250   =\z@
251 \def\saved@space@catcode{10}
252 \let\in@penc@test\relax
253 \def\IeC{%
254   \ifx\protect\@typeset@protect
255     \expandafter\@firstofone
256   \else
257     \noexpand\IeC
258   \fi
259 }
```

Make characters active for UTF-8 input formats

```

260 \@tempcnta=1
261 \loop
262   \catcode\@tempcnta=13 %
263   \advance\@tempcnta\@ne %
264   \ifnum\@tempcnta<32 %
265   \repeat %
266 \catcode0=15 % null
267 \catcode9=10 % tab
268 \catcode10=12 % ctrl J
269 \catcode12=13 % ctrl L
270 \catcode13=5 % newline
271 \@tempcnta=128
272 \loop
273   \catcode\@tempcnta=13
274   \advance\@tempcnta\@ne
275   \ifnum\@tempcnta<256
276   \repeat

```

`\UseRawInputEncoding` Reset 8 bit characters to catcode 12 so the input encoding matches the “Raw” font encoding. Useful for special behaviours, or for compatibility with older L^AT_EX formats.

```

277 \def\UseRawInputEncoding{%
278 \let\inputencodingname\undefined % revert
279 \let\DeclareFontEncoding\DeclareFontEncoding@saved % revert
280 \let\DeclareUnicodeCharacter\undefined % revert
281 \@tempcnta=1
282 \loop
283   \catcode\@tempcnta=15 %
284   \advance\@tempcnta\@ne %
285   \ifnum\@tempcnta<32 %
286   \repeat %
287 \catcode0=15 % null
288 \catcode9=10 % tab
289 \catcode10=12 % ctrl J
290 \catcode12=13 % ctrl L
291 \catcode13=5 % newline
292 \@tempcnta=128
293 \loop
294   \catcode\@tempcnta=12
295   \advance\@tempcnta\@ne
296   \ifnum\@tempcnta<256
297   \repeat
298 }

```

`\DeclareFontEncoding@saved` Saved version of `\DeclareFontEncoding@` before `utf8.def` modifies it for use in `\UseRawInputEncoding` above.

```

299 \let\DeclareFontEncoding@saved\DeclareFontEncoding@

300 \edef\inputencodingname{utf8}%
301 \input{utf8.def}
302 \let\UTFviii@undefined@err@@\UTFviii@undefined@err
303 \let\UTFviii@invalid@err@@\UTFviii@invalid@err

```

```

304 \let\UTFviii@two@octets@@\UTFviii@two@octets
305 \let\UTFviii@three@octets@@\UTFviii@three@octets
306 \let\UTFviii@four@octets@@\UTFviii@four@octets
307 (2ekernel)\def\UTFviii@undefined@err#1{\@gobble#1}%
308 (2ekernel)\let\UTFviii@invalid@err\string
309 (2ekernel)\let\UTFviii@two@octets\string
310 (2ekernel)\let\UTFviii@three@octets\string
311 (2ekernel)\let\UTFviii@four@octets\string
312 (2ekernel)\everyjob\expandafter{\the\everyjob
313 (2ekernel)\let\UTFviii@undefined@err\UTFviii@undefined@err@@
314 (2ekernel)\let\UTFviii@invalid@err\UTFviii@invalid@err@@
315 (2ekernel)\let\UTFviii@two@octets\UTFviii@two@octets@@
316 (2ekernel)\let\UTFviii@three@octets\UTFviii@three@octets@@
317 (2ekernel)\let\UTFviii@four@octets\UTFviii@four@octets@@
318 (2ekernel)}
319 \let\@inpenc@test\@undefined
320 \let\saved@space@catcode\@undefined

```

For formats not set up for UTF-8 default, set the C0 controls to catcode 15.

```

321 \else
322 \@tempcnta=0
323 \loop
324   \catcode\@tempcnta=15 %
325   \advance\@tempcnta\@ne %
326   \ifnum\@tempcnta<32 %
327     \repeat %
328 \catcode0=15 % null
329 \catcode9=10 % tab
330 \catcode10=12 % ctrl J
331 \catcode12=13 % ctrl L
332 \catcode13=5 % newline
333 \let\UseRawInputEncoding\relax

```

This ends the skipped code in Unicode engines:

```

334 \fi
335 (/2ekernel|latexrelease)
336 (latexrelease)\EndIncludeInRelease
337 (latexrelease)\IncludeInRelease{0000/00/00}%
338 (latexrelease)      {\UTFviii@invalid}{UTF-8 default}%

```

The first block of commands got only introduced in 2019 but we revert all of Unicode support in one go not jump to the intermediate version.

```

339 (latexrelease) \let\UTFviii@two@octets@combine\@undefined
340 (latexrelease) \let\UTFviii@three@octets@combine\@undefined
341 (latexrelease) \let\UTFviii@four@octets@combine\@undefined
342 (latexrelease) \let\UTFviii@two@octets@string\@undefined
343 (latexrelease) \let\UTFviii@three@octets@string\@undefined
344 (latexrelease) \let\UTFviii@four@octets@string\@undefined
345 (latexrelease) \let\UTFviii@two@octets@noexpand\@undefined
346 (latexrelease) \let\UTFviii@three@octets@noexpand\@undefined
347 (latexrelease) \let\UTFviii@four@octets@noexpand\@undefined
348 (latexrelease)\@tempcnta=0
349 (latexrelease)\loop
350 (latexrelease) \catcode\@tempcnta=15

```



```

351 \latexrelease \advance\@tempcnta\@ne
352 \latexrelease \ifnum\@tempcnta<32
353 \latexrelease \repeat %
354 \latexrelease \catcode9=10 % tab
355 \latexrelease \catcode10=12 % ctrl J
356 \latexrelease \catcode12=13 % ctrl L
357 \latexrelease \catcode13=5 % newline
358 \latexrelease \@tempcnta=128
359 \latexrelease \loop
360 \latexrelease \catcode\@tempcnta=12
361 \latexrelease \advance\@tempcnta\@ne
362 \latexrelease \ifnum\@tempcnta<256
363 \latexrelease \repeat
364 \latexrelease \let\IeC\@undefined
365 \latexrelease \def\DeclareFontEncoding@#1#2#3{%
366 \latexrelease \expandafter
367 \latexrelease \ifx\csname T@#1\endcsname\relax
368 \latexrelease \def\cdp@elt{\noexpand\cdp@elt}%
369 \latexrelease \xdef\cdp@list{\cdp@list\cdp@elt{#1}}%
370 \latexrelease {\default@family}{\default@series}%
371 \latexrelease {\default@shape}}%
372 \latexrelease \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
373 \latexrelease \else
374 \latexrelease \@font@info{Redeclaring font encoding #1}%
375 \latexrelease \fi
376 \latexrelease \global\@namedef{T@#1}{#2}%
377 \latexrelease \global\@namedef{M@#1}{\default@M#3}%
378 \latexrelease \xdef\LastDeclaredEncoding{#1}%
379 \latexrelease }
380 \latexrelease \let\UseRawInputEncoding\@undefined
381 \latexrelease \let\DeclareFontEncoding@saved\@undefined
382 \latexrelease \let\inputencodingname\@undefined
383 \latexrelease \EndIncludeInRelease

384 \*2kernel)
385 % \begin{macrocode}
386 %
387 % We temporarily define |\reserved@a| to apply |\reserved@c| to all the
388 % numbers in the range of its arguments.
389 % \begin{macrocode}
390 \def\reserved@a#1#2{%
391 \@tempcnta#1\relax
392 \@tempcntb#2\relax
393 \reserved@b
394 }
395 \def\reserved@b{%
396 \ifnum\@tempcnta>\@tempcntb\else
397 \reserved@c\@tempcnta
398 \advance\@tempcnta\@ne
399 \expandafter\reserved@b
400 \fi
401 }

```

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.

```
402 \catcode'\ =10
403 \catcode'\#=6
404 \catcode'\$=3
405 \catcode'\%=14
406 \catcode'\&=4
407 \catcode'\\=0
408 \catcode'\^=7
409 \catcode'\_ =8
410 \catcode'\{=1
411 \catcode'\}=2
412 \catcode'\~=13
413 \catcode'\@=11
414 \catcode'\^^I=10
415 \catcode'\^^J=12
416 \catcode'\^^L=13
417 \catcode'\^^M=5
```

Set the ‘other’ catcodes.

```
418 \def\reserved@c#1{\catcode#1=12\relax}
419 \reserved@c{'\!}
420 \reserved@c{'\"}
421 \reserved@a{'\' }{'\?}
422 \reserved@c{'\[}
423 \reserved@c{'\]}
424 \reserved@c{'\' }
425 \reserved@c{'\|}
```

Set the ‘letter’ catcodes.

```
426 \def\reserved@c#1{\catcode#1=11\relax}
427 \reserved@a{'\A}{'\Z}
428 \reserved@a{'\a}{'\z}
```

All the characters in the range 0–31 and 127–255 are illegal, *except* tab (\textasciitilde I), nl (\textasciitilde J), ff (\textasciitilde L) and cr (\textasciitilde M).

88.7 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.

```
429 \ifnum 0%
430   \ifx\Umathcode\@undefined\else 1\fi
431   \ifx\XeTeXmathcode\@undefined\else 1\fi
432   >\z@
433 \else
434 \def\reserved@c#1{%
435   \count@=#1\advance\count@ by -"20
436   \uccode#1=\count@
437   \lccode#1=#1
438 }
```

```

439 \reserved@a{'\a}{'\z}
440 \reserved@a{"A0}{\BC}
441 \reserved@a{"E0}{\FF}

```

The upper case characters need their \uccode and \lccode values set, and their \sfcode set to 999.

```

442 \def\reserved@c#1{%
443   \count@=#1\advance\count@ by "20
444   \uccode#1=#1
445   \lccode#1=\count@
446   \sfcode#1=999
447 }
448 \reserved@a{'\A}{'\Z}
449 \reserved@a{"80}{\9C}
450 \reserved@a{"C0}{\DF}

```

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.

```

451 \uccode'\^^Y='I      % dotless i
452 \lccode'\^^Y='^^Y    % dotless i
453 \uccode'\^^Z='J      % dotless j, ae in OT1
454 \lccode'\^^Z='^^Z    % dotless j, ae in OT1
455 \lccode'\^^9d='i     % dotted I
456 \uccode'\^^9d='^^9d % dotted I
457 \lccode'\^^9e='^^9e % d-bar
458 \uccode'\^^9e='^^d0 % d-bar

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

459 \lccode'\^^[='^^[    % oe in OT1
460 \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.

```

461 \DeclareRobustCommand{\MakeUppercase}[1]{%
462   \def\i{I}\def\j{J}%
463   \def\reserved@a##1##2{\let##1##2\reserved@a}%
464   \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%

```

Tell UTF-8 processing to process chars even though we are in an \protected@edef.

```

465   \let\UTF@two@octets@noexpand\@empty
466   \let\UTF@three@octets@noexpand\@empty
467   \let\UTF@four@octets@noexpand\@empty
468   \protected@edef\reserved@a{\uppercase{#1}}%
469   \reserved@a
470 }
471 \DeclareRobustCommand{\MakeLowercase}[1]{%
472   \def\reserved@a##1##2{\let##2##1\reserved@a}%
473   \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
474   \let\UTF@two@octets@noexpand\@empty

```

```

475 \let\UTF@three@octets@noexpand\@empty
476 \let\UTF@four@octets@noexpand\@empty
477 \protected@edef\reserved@a{\lowercase{#1}}%
478 \reserved@a
479 }}
480 \def\@uclclist{\oe\OE\o\O\ae\AE
481 \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:

```

482 \protected@edef\MakeUppercase#1{\MakeUppercase{#1}}
483 \protected@edef\MakeLowercase#1{\MakeLowercase{#1}}

```

88.8 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.

```

484 %\IfFileExists{ltpatch.ltx}
485 % {\typeout{=====^~J%
486 %      Applying patch file ltpatch.ltx^~J%
487 %      =====}}
488 % \def\fmtversion@topatch{unknown}
489 % \input{ltpatch.ltx}
490 % \ifx\fmtversion\fmtversion@topatch
491 % \ifx\patch@level\@undefined
492 % \typeout{^~J^~J^~J%
493 %      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^~J%
494 %      !! Patch file 'ltpatch.ltx' not suitable for this^~J%
495 %      !! version of LaTeX.^~J^~J%
496 %      !! Please check if initex found an old patch file:^~J%
497 %      !! --- if so, rename it or delete it, and redo the^~J%
498 %      !! initex run.^~J%
499 %      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^~J}%
500 % \batchmode \@@end
501 % \else

```

The code below adds the 'patch level' string to the first `\typeout` in the startup banner.

```

502 % \def\fmtversion@topatch{0}%

```


88.11 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`.

```
538 \def\@providesfile#1[#2]{%
539     \wlog{File: #1 #2}%
540     \expandafter\xdef\csname ver@#1\endcsname{#2}%
541     \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 \LaTeX 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.

```
542 \let\@filelist\@gobble
543 \def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}%
```

88.12 Some last minute initializations ...

This initializes the 2020/02/02 extensions to NFSS after any changes in the preamble.

```
544 \AtBeginDocument{\reinstall@nfss@defs\init@series@setup}
```

88.13 Do some temporary work for pre-release

This is a good place to load code that hasn't yet been integrated into the other files ...

```
545 %\input ...
```

88.14 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.

```
546 \makeatother
547 \errorstopmode
548 \dump
549 </2ekernel>
```

Change History

- 1985-11-04 ltmath.dtx LaTeX2.09
 General: produce warning message if line extends into margin.
 Doesn't warn about formula overprinting equation number. 359
- 1989-04-10 ltffssbas.dtx v1.0a
 General: Starting with version numbers! `\ifmmode` added in `\math@group` 153, 176
- 1989-04-10 ltffssbas.dtx v1.0b
 General: `\preload@sizes` added. 153, 176
`\wrong@fontshape` changed to define substitution font/shape macro. 153, 176
- 1989-04-10 ltffssini.dtx v1.0a
 General: Starting with version numbers `\newif` for `\@tempswa` added since this switch is unknown at the time when this file is read in. (latex.tex is loaded later.) `\math@famname` changed to `\math@version`. . 252
- 1989-04-14 ltffssbas.dtx v1.0c
 General: More documentation added. 153, 176
- 1989-04-15 ltffssini.dtx v1.0b
 General: `\mathfontset` renamed to `\mathversion`. 252
- 1989-04-19 ltffssbas.dtx v1.0d
 General: Even more doc. . . 153, 176
- 1989-04-21 ltffssbas.dtx v1.0e
 General: Documentation is fun!
 Parameters of `\define@mathalphabet` changed. 153, 176
- 1989-04-21 ltffssini.dtx v1.0c
 General: Changed to conform to fam.tex. 252
- 1989-04-23 ltffssbas.dtx v1.0f
 General: % in `\getanddefinefonts` added. 153, 176
- 1989-04-26 ltffssini.dtx v1.0d
 General: `\xpt` added. 252
- 1989-04-27 ltffssbas.dtx v1.0g
 General: Documentation revised. 153, 176
- 1989-04-27 ltffssini.dtx v1.0e
 General: Definitions of L^AT_EX symbols corrected. 252
- 1989-04-29 ltffssbas.dtx v1.0h
 General: Documented problem with `\halign`, and `\noalign` 153, 176
`\mathversion`: Test if version defined added. 162, 185
- 1989-04-29 ltffssbas.dtx v1.0i
 General: Removed the `\halign` `\noalign` correction (wasn't bugfree) 153, 176
- 1989-04-29 ltffssini.dtx v1.0f
 General: Corrections to L^AT_EX tabular env. added. 252
- 1989-05-01 ltffssbas.dtx v1.0j
 General: Default for `\baselinestretch` added. 153, 176
- 1989-05-22 ltffssbas.dtx v1.0k
 General: Lines longer than 72 characters folded. 153, 176
- 1989-05-22 ltffssini.dtx v1.0g
 General: Lines shortened to 72 characters 252
- 1989-09-14 ltffssbas.dtx v1.0m
 General: Global replacement: `\group` to `\mathgroup` . . 153, 176
`\mathversion`: Corrected typo: `\endscname` to `\endcsname`. 162, 185
- 1989-11-07 ltffssini.dtx v1.0i
 General: All family, series, and shape names abbreviated. . . 252
- 1989-11-08 ltffssbas.dtx v1.0o
 General: First parameter of `\define@mathalphabet` and `\define@mathgroup` changed from string to control sequence. 153, 176
- 1989-11-14 ltffssbas.dtx v1.0p
`\math@version`: Math version prefix 'mv@' added. . . . 162, 185
- 1989-11-19 ltffssbas.dtx v1.0q
`\define@newfont`: Group added. 163, 186
`\wrong@fontshape`: Instead of calling `\family\default@family`, etc.

- we directly set `\f@family`,
etc. 168, 191
- 1989-11-22 ltfssbas.dtx v1.0r
 `\math@version: \def → \edef` for
 `\math@version`. 162, 185
- 1989-11-25 ltfssbas.dtx v1.0s
 General: All `\edef\font@name`
 changed to `\xdef\font@name`.
 Necessary after introduction of
 `\begingroup/\endgroup` in
 v1.0q. 153, 176
 extra// → + in
 `\extra@def`. 153, 176
- 1989-11-26 ltfssbas.dtx v1.0t
 `\select@group: \bgroup/\egroup`
 changed to
 `\begingroup/\endgroup` to
 avoid empty Ord atom on
 math list. 170, 193
- 1989-12-02 ltfssini.dtx v1.1b
 General: `\rmmath` renamed to
 `\mathrm`. 252
- 1989-12-03 ltfssini.dtx v1.1c
 General: Some internal macros
 renamed to make them
 inaccessible. 252
- 1989-12-05 ltfssbas.dtx v1.0u
 `\addto@hook: \addto@hook`
 added. 175, 198
- 1989-12-05 ltfssstrc.dtx v1.0u fam.dtx
 `\every@math@size: Hook`
 `\every@size` added. 206
- 1989-12-13 ltfssstrc.dtx v1.0f
 `\use@mathgroup: \expandafter`
 added before final `\fi`. . . . 209
- 1989-12-16 ltfssbas.dtx v1.1a
 `\select@group: \relax` in front
 added. 170, 193
 Now four arguments. . . . 170, 193
 Redefinition of alphabet now
 simpler. 171, 194
 Usage of ‘=’ macro added. 171, 194
- 1989-12-16 ltfssstrc.dtx v1.1a
 `\selectfont: Changed order of`
 calls. 203
 `\use@mathgroup: Redefinition of`
 alphabet now simpler. 209
 Usage of ‘=’ macro added. . . . 209
- 1990-01-18 ltfssstrc.dtx v1.0h
 General: `\tracingfonts` meaning
 changed. 199
- 1990-01-20 ltfssbas.dtx v1.2a
 `\math@bgroup: Def. placed in this`
 file. 172, 195
- `\math@egroup: Def. placed in this`
 file. 172, 195
- `\select@group: Def for alph id`
 changed. 171, 194
- 1990-01-21 ltfssbas.dtx v1.2b
 `\select@group: Code moved to`
 `\use@mathgroup`. 171, 194
- 1990-01-21 ltfssstrc.dtx v1.2b
 `\use@mathgroup: Macro added to`
 allow cleaner interface. 209
- 1990-01-23 ltfssbas.dtx v1.2c
 General: `\no@version@warning`
 renamed to
 `\no@alphabet@error`. . . 153, 176
 Macro `\no@alphabet@help`
 added 153, 176
 `\no@alphabet@error: Changed to`
 error call 153, 176
- 1990-01-25 ltfssini.dtx v1.1e
 `\nfss@text: Macro added`. 265
- 1990-01-27 ltfssbas.dtx v1.2d
 `\DeclarePreloadSizes: Font`
 identifier set to `\relax`. . 158, 181
- 1990-01-28 ltfssbas.dtx v1.2e
 `\mathgroup: \newfam` let to
 `\new@mathgroup`. 153, 176
- 1990-01-28 ltfssbas.dtx v1.2f
 `\define@newfont: Added call to`
 `\curr@fontshape` macro to
 allow substitution. 164, 187
- `\wrong@fontshape: Warning`
 message slightly changed. 168, 191
- 1990-01-28 ltfssini.dtx v1.2b
 `\em: Call to \@nomath` added. . . . 263
- 1990-02-08 ltfssini.dtx v1.1g
 General: Protected the commands
 `\family`, `\series`, `\shape`,
 `\size`, `\selectfont`, and
 `\mathversion`. 252
- 1990-02-16 ltfssbas.dtx v1.2g
 General: Support for changes of
 `\baselineskip` without
 changing the size. 153, 176
- `\math@version: \@nomath`
 added. 162, 185
- 1990-02-16 ltfssstrc.dtx v1.0i
 `\selectfont: Changed \f@size to`
 `\lcl@currsize` (see fam file). 203
- 1990-02-18 ltfssstrc.dtx v1.0j
 General: Redefine unprotected
 version `\p@selectfont` instead
 of `\selectfont`. 203
- 1990-03-14 ltfssstrc.dtx v1.0k
 General: Added code for TeX3. . . 199

<code>\extract@font</code> : Added code for TeX3.	202	1990-07-07 ltfssstrc.dtx v1.0q	<code>\getanddefine@fonts</code> : Group number added to tracing. . .	210
<code>\selectfont</code> : Added code for TeX3.	203	<code>\math@egroup</code> : Tracing code added.	209	
1990-03-30 ltfssbas.dtx v1.2h		<code>\use@mathgroup</code> : Group number added to tracing.	209	
<code>\math@egroup</code> : Changed to have one arg.	172, 195	1990-08-27 ltfssstrc.dtx 1.0r	<code>\type@restoreinfo</code> : Some extra tracing info.	205
1990-03-30 ltfssstrc.dtx v1.2h		<code>\getanddefine@fonts</code> : Correcting missing name after		
<code>\use@mathgroup</code> : Third argument removed (see <code>\math@egroup</code>). . .	209	<code>\tracingon</code>	210	
1990-04-01 ltfssbas.dtx v1.2i		1991-03-28 ltfssini.dtx v1.1m	<code>\copyright</code> : Extra braces added. . .	265
General: Code added from tracefnt.dtx.	153, 176	1991-03-30 ltfssini.dtx v1.2g	<code>\newfont</code> : Definition added. . . .	264
Support for TeX3.	153, 176	<code>\symbol</code> : Definition added.	264	
1990-04-01 ltfssstrc.dtx v1.0l		1991-07-24 ltmiscen.dtx LaTeX2.09	<code>\@verbatim</code> : Added	
General: Part of code moved to fam.dtx.	199	<code>\penalty\interlinepenalty</code> to definition of <code>\par</code> so that		
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> already defined.	200	<code>\samepage</code> works	346	
1990-04-01 ltfssstrc.dtx v1.0o		1991-08-14 ltmath.dtx LaTeX2.09	<code>\cases</code> : (RmS) inserted extra braces around entry for NFSS	355
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> defined removed again.	200	1991-08-14 ltpictur.dtx LaTeX2.09	General: (RmS) inserted extra braces around entry for NFSS	417
1990-04-02 ltfssini.dtx v1.1i		1991-08-14 ltthm.dtx LaTeX2.09	<code>\@endtheorem</code> : Moved <code>\itshape</code> after <code>\item</code> to make it work with NFSS	440
General: <code>\input</code> of files now handled by docstrip.	252	1991-08-26 ltfssini.dtx v1.1n	<code>\p@reset@font</code> : Macro introduced	265
1990-04-05 ltfssstrc.dtx v1.0m		1991-08-26 ltmiscen.dtx LaTeX2.09	<code>\@verbatim</code> : <code>\@par</code> added	346
<code>\selectfont</code> : Call <code>\tracingon</code> only if <code>\tracingfonts</code> greater than 3.	203	1991-08-26 ltpictur.dtx LaTeX2.09	<code>\endpicture</code> : (RmS & FMi) extra boxing level around <code>\@picbox</code> to guard against unboxing in math mode (proposed by John Hobby)	416
1990-05-05 ltfssstrc.dtx v1.0n		1991-08-26 ltplain.dtx LaTeX2.09	<code>\tracingall</code> : Added <code>\errorcon-</code> <code>textlines=\maxdimen</code> , suggested by J. Schrod	30
<code>\selectfont</code> : <code>\tracingon</code> with new syntax.	203	1991-09-29 ltboxes.dtx LaTeX2.09	<code>\mpfootnotetext</code> : (RmS) added	
1990-06-23 ltfssini.dtx v1.1k		<code>\reset@font</code>	389	
<code>\nfss@text</code> : Changed to <code>\mbox</code> . . .	265	1991-09-29 ltfloa.dtx LaTeX2.09	<code>\@footnotetext</code> : (RmS) added	
1990-06-24 ltfssbas.dtx v1.2j		<code>\reset@font</code>	470	
<code>\DeclarePreloadSizes</code> : Missing percent added.	157, 180			
1990-06-24 ltfssstrc.dtx v1.0o				
<code>\baselinestretch</code> : Moved to tracefnt.dtx.	206			
<code>\getanddefine@fonts</code> : <code>\Adding</code> tracing code.	210			
<code>\Macro</code> moved from fam.dtx. . .	210			
Adding debug code.	210			
<code>\use@mathgroup</code> : Tracing code added.	209			
1990-06-30 ltfssbas.dtx v1.2l				
<code>\showhyphens</code> : Macro added. . .	173, 196			
1990-06-30 ltfssstrc.dtx v1.0p				
<code>\use@mathgroup</code> : Added <code>\relax</code> after math group number. . .	209			

- 1991-09-29 ltmath.dtx LaTeX2.09
`\@eqnnum`: RmS: `\reset@font`
added. 358
- 1991-09-29 ltsect.dtx LaTeX2.09
`\@dottedtocline`: (RmS) added
`\reset@font` for page number 451
- 1991-10-17 ltcntrl.dtx LaTeX2.09
`\@tfor`: (RmS) `\xdef` replaced by
`\def` (See FMI's array.doc) . . . 59
- 1991-10-25 ltbibl.dtx LaTeX2.09
`\@citex`: added `\reset@font`,
suggested by Bernd Raichle. 475
- 1991-11-01 ltfloat.dtx LaTeX2.09
`\footnote`: (RmS) Added
`\let\protect\noexpand` in
`\footnote`, `\footnotemark`,
and `\footnotetext`, since
`\xdef` is used 470
- 1991-11-04 ltlists.dtx LaTeX2.09
`\makelabel`: (RmS) added default
definition for `\makelabel`, to
produce an error message. . . 376
- 1991-11-04 ltplain.dtx RmS
General: Removed `\itemitem` since
never needed/useful with
 \LaTeX 28
- 1991-11-06 ltbibl.dtx LaTeX2.09
`\@citex`: added code to remove a
leading blank 475
- 1991-11-13 ltbibl.dtx LaTeX2.09
`\@bibitem`: Changed counter
`enumi` to `enumiv`, as it says in
the comment above 475
- 1991-11-21 ltffssini.dtx v1.1o
`\p@reset@font`: Added extra
braces for robustness. 265
Changed to protected version of
macro. 265
- 1991-11-22 ltfloat.dtx LaTeX2.09
`\footnote`: (RmS) Added
`\let\protect\noexpand` in
`\@xfootnote`,
`\@xfootnotemark`, and
`\@xfootnotetext` 470
- 1991-11-22 ltlists.dtx LaTeX2.09
`\@item`: (RmS) Changed second
call to `\makelabel` to
`\unhbox\@tempboxa`. Avoids
problems with side effects in
`\makelabel` and is more
efficient. 376
- 1991-11-27 ltffssbas.dtx v1.3a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily`
etc. 153, 176
- 1991-11-27 ltffssini.dtx v1.2a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily` etc. 252
- 1992-01-06 ltffssini.dtx v1.2c
General: added `slitex` code 252
- 1992-01-10 ltbibl.dtx LaTeX2.09
`\@bibitem`: Changed `\c@enumiv` to
`\value of \@listctr` 475
- 1992-01-10 ltmath.dtx LaTeX2.09
`equation`: RmS: put `\hbox` around
`\@eqnnum` to typeset the
equation number in text mode
(as in the `eqnarray` env.) . . . 358
- 1992-01-10 ltthm.dtx LaTeX2.09
`\@othm`: (RmS) Check for existence
of theorem environment . . . 439
- 1992-01-14 ltbibl.dtx LaTeX2.09
`\@biblabel`: removed `\hfill` . . . 477
- 1992-01-14 ltsect.dtx 0.0
`\@starttoc`: (RmS) added
`\immediate` to `\openout` as all
`\write` commands are also
executed `\immediate` 449
- 1992-02-26 ltbibl.dtx LaTeX2.09
`\@lbibitem`: Added `\hfill` to
restore left-alignment of
bibliography labels in alpha
style 475
- 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` 37
- 1992-03-18 ltfloat.dtx LaTeX2.09
`\@xympar`: (RmS) added
`\global\@ignorefalse` 466
- `\end@float`: (RmS) changed
`\@esphack` to `\@Esphack` . . . 460
- 1992-03-18 ltlists.dtx 0.0
General: RmS: added
`\@nmbulistfalse` 373
- 1992-03-18 ltmiscen.dtx LaTeX2.09
`\begin`: Changed `\@ignoretrue` to
`\@ignorefalse` (as
documented) 341
- 1992-03-21 ltffssini.dtx v1.2d
General: Renamed `\text` to
`\nfss@text` to make it
internal. 252
- 1992-05-12 ltffssbas.dtx v1.3c
`\extract@alph@from@version`:
Macro added. 171, 194

- `\select@group`: Added call to `\extract@alph@from@version`. 171, 194
- 1992-07-26 `ltfssbas.dtx` v1.9a
 - `\curr@fontshape`: 163, 186
 - `\DeclareFontShape`: Introduced
 - `\DeclareFontShape` . . . 154, 177
 - `\define@newfont`: 163, 186
 - `\math@fonts`: 170, 193
 - `\select@group`: 171, 194
 - `\split@name`: Added splitting into
 - `\f@encoding`. 163, 186
 - `\wrong@fontshape`: 168, 191
- 1992-07-26 `ltfsstrc.dtx` v2.0b
 - `\s@fct@`: 218
 - `\selectfont`: 203
 - `\try@simple@size`: 212, 213
 - `\try@size@range`: 216
 - `\use@mathgroup`: 209
- 1992-08-14 `ltbibl.dtx` LaTeX2.09
 - `\citex`: added missing argument braces around `\hbox`, found by Ed Sznyter 475
- 1992-08-14 `ltboxes.dtx` LaTeX2.09
 - `\endminipage`: (RmS) replaced `\vskip`-`\lastskip` by `\unskip` (proposed by FMI) 388
- 1992-08-17 `ltbibl.dtx` LaTeX2.09
 - `\citex`: simplified code for removing leading blanks in citation key (proposed by Frank Jensen and Kresten Krab Thorup) 475
- 1992-08-19 `ltsect.dtx` 0.0
 - `\@xsect`: (RmS) corrected bug: stretch and shrink in argument to `\hskip` previously not negated 446
- 1992-08-19 `ltthm.dtx` LaTeX2.09
 - `\@othm`: (RmS) Changed error message to complain about undefined counter 439
- 1992-08-20 `ltfssini.dtx` v1.4b
 - `\@setsize`: Added `\@currsiz`. . . 265
- 1992-08-24 `ltdefns.dtx` LaTeX2.09
 - `\@ifnextchar`: (Rms)
 - `\@ifnextchar` didn't work if its first argument was an equal sign. 50
- 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`. . . . 341
- `\verb`: Changed `\verb` and `\@sverb` to work correctly in math mode 349
- 1992-08-25 `ltsect.dtx` LaTeX2.09
 - `\@sect`: (FMI) replaced explicit setting of `\@svsec` by call to `\@secCNTformat` 445
- 1992-09-18 `ltlists.dtx` LaTeX2.09
 - `\item`: (RmS) Added warning if `\item` is used in math mode . . 374
- 1992-09-18 `lftab.dtx` LaTeX2.09
 - `\@array`: Changed `\par` to `\@empty` to avoid starting new row e.g. after `\hline` 404
- 1992-09-19 `ltfsstrc.dtx` v2.0c
 - `\try@simple@size`: 212
- 1992-09-21 `ltfssini.dtx` v1.4d
 - `\not@math@alphabet`: Macro defined. 264
- 1992-09-22 `ltfssbas.dtx` v1.91a
 - General: Introduced `\tf@size` for math size. 153, 176
- 1992-09-22 `ltfsstrc.dtx` v2.1a
 - `\getanddefine@fonts`: Introduced `\tf@size` for math size. . . . 210
- 1992-11-13 `ltfssini.dtx` v?
 - `\hexnumber@`: Made expandable. . 265
- 1992-11-23 `ltcounts.dtx` LaTeX2.09
 - `\stepcounter`: Replaced `{}` in `\stepcounter` by `\begingroup` `\endgroup` to avoid adding an empty ord in math mode . . . 146
- 1992-11-26 `ltboxes.dtx` LaTeX2.09
 - `\@mpfootnotetext`: (RmS) added protection for `\edef` 389
- 1992-11-26 `ltfloat.dtx` LaTeX2.09
 - `\@footnotetext`: (RmS) added protection for `\edef` 470
- `\footnote`: (RmS) Changed all to `'def'protect'noexpand'protect'noexpand` 470
- 1992-12-03 `ltfssini.dtx` v?
 - `\hexnumber@`: Make it accept counters. 265
- 1993-03-08 `preload.dtx` v2.0b
 - General: Added 12pt preloads . . 288
- 1993-03-18 `ltfssbas.dtx` v2.0c
 - General: Changed all `\@tempdima` in `\@tempdimb` to avoid killing `\numberline` 153, 176
- 1993-03-18 `ltfsstrc.dtx` v2.1b
 - General: Changed all `\@tempdima` in `\@tempdimb` to avoid killing `\numberline` 199

- Changed all `\@tempdimb` in
`\@tempdimx` to avoid killing
`\numberline` 199
- 1993-03-18 `ltfsstrc.dtx` v2.1c
`\DeclareSizeFunction`: Added all
args to avoid blanks problems 215
- 1993-04-09 `lterror.dtx` v1.0e
`\@latexerr`: Mention The
Companion 65
- 1993-04-11 `lterror.dtx` v1.0f
`\@latexerr`: Remove setting of
`errorcontextlines` 65
- 1993-05-05 `ltfntcmd.dtx` v2.0b
General: Removed all LaTeX
related cmds 292
- 1993-05-16 `ltfssbas.dtx` v2.0e
`\showhyphens`: Use
`\reset@font` 173, 196
- 1993-07-16 `ltfsstrc.dtx` v2.1h
General: Changed layout of info
messages 199
- 1993-07-17 `ltoutenc.dtx` 1.0d
General: changed `\catcoding @` . 105
- 1993-08-03 `ltmiscen.dtx` LaTeX2.09
`\enddocument`: Changed
redefinition of `\global` to
redefinition of `\setckpt`. . . 337
- 1993-08-05 `ltpictur.dtx` LaTeX2.09
`\circle`: (RMS) Added error
message if `\circle` is used in
math mode. 432
- 1993-08-05 `ltsect.dtx` LaTeX2.09
`\@sect`: (RmS) Made sure that
`\protect` works correctly in
expansion of `\the` counter . . 445
- 1993-08-05 `ltspace.dtx` LaTeX2e
`\@hspace`: (RmS) Removed
superfluous `\leavevmode` in
`\@hspace` and `\@hspacer`, as
suggested by CAR. 85
- 1993-08-05 `lftab.dtx` latex2e
`\tabular*`: Replaced
`\expandafter\def` by
`\@namedef`. 403
- 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. 475
- 1993-08-13 `ltoutenc.dtx` 1.0f
General: Protected against active
`@` sign. 105
- 1993-08-13 `preload.dtx` v2.0c
General: Added `\relax` at end of
font names. 289
- 1993-08-16 `ltoutenc.dtx` 1.0g
General: Needs space after
`\string` 105
- 1993-08-18 `ltfssdcl.dtx` v2.0e
`\new@mathversion`: Exchanged
names of encodings in warning
message of `\SetSymbolFont`. 234
- 1993-09-02 `ltfsstrc.dtx` v2.1i
General: Corrected name of `sgen`
size function. 199
- 1993-09-03 `ltmiscen.dtx` LaTeX2.09
`\verbatim@nolig@list`: Replaced
`\@noligs` by extensible list . 350
- 1993-09-07 `ltmiscen.dtx` LaTeX2.09
`\verb@balance@group`: (RmS)
Changed definition of `\verb` so
that it detects a missing second
delimiter. 349
- 1993-09-08 `ltmiscen.dtx` LaTeX2.09
`\enddocument`: Added warning in
case of undefined references. 337
- 1993-09-15 `ltfssbas.dtx` v2.0g
`\DeclareFontEncoding`: Corrected:
`\default@T` to
`\default@M`. 156, 179
- 1993-09-15 `ltfsstrc.dtx` v2.1j
General: Corrected spelling of
`\noexpand`. 199
- 1993-09-19 `lterror.dtx` LaTeX2.09
`\@invalidchar`: (RmS) Error
message for invalid input
characters. 67
- 1993-11-02 `ltmath.dtx` LaTeX2.09
General: RmS: Corrected
description of `\@eqnset`,
moved `\@eqnset` accordingly
and removed extra `\tabskip`
assignment. 359
- 1993-11-03 `ltmath.dtx` LaTeX2e
General: RmS: Initialized
`\everycr` to empty 359
- 1993-11-03 `ltpictur.dtx` LaTeX2.09
General: (RmS) changed `\halign`
to `\ialign` to initialize
`\tabskip` and `\everycr` . . . 417
- 1993-11-11 `ltfssini.dtx` v2.1a
`\normalfont`: Macro added . . . 265
- 1993-11-11 `ltfsstrc.dtx` v2.2a
General: Option concept added for
LaTeX2e 199

1993-11-14 ltclass.dtx v0.2a	1993-11-21 ltfiles.dtx LaTeX2e
\@current: Name changed from	\@missingfileerror: Stop infinite
\@currentextension 558	looping on \@er@ext 99
\@fileswithoptions: Moved	1993-11-21 ltmiscen.dtx v0.9a
resetting of \default@ds, \ds@	\@verbatim: use \@verbatim@font
and \@declaredoptions here,	instead of \tt 347
from the end of	\verb: Use \@verbatim@font
\ProcessOptions. 568	instead of \tt. 349
\@resetOptions: macro added . . 570	\verbatim@font: Macro added . . 347
\AtEndDocument: Included	1993-11-22 ltclass.dtx v0.2f
extension in the generated	\@fileswithoptions: Made the
macro name for package and	default [] not
class hooks. 570	[\@unknownversion] 567
\documentstyle: Added	Made the initial version [] not
\RequirePackage	[\@unknownversion] 568
\@unusedoptionlist stuff. . . 565	\@ifclasslater: Added //00 so
\g@addto@macro: Made global . . 570	parsing never produces a
\NeedsTeXFormat: made more	runaway argument. 559
robust for alternative syntax	General: \@unknownversion
for other formats. 566	removed 553
\ProcessOptions*: Optimise	1993-11-22 ltdefns.dtx LaTeX2e
‘empty option’ code. 563	\@minus: Macro added 36
Stop adding the global option	\@plus: Macro added 36
list inside class files. 563	\CheckCommand: Macro added . . . 43
1993-11-15 ltclass.dtx v0.2b	\providecommand: Macro added . . 43
\documentstyle: Modified to	1993-11-22 lterror.dtx LaTeX2e
match \ProcessOption* . . . 565	\c@errorcontextlines: Macro
\ProcessOptions*: Star form	added 64
added. 563	1993-11-22 ltfiles.dtx LaTeX2e
1993-11-17 ltclass.dtx v0.2c	\listfiles: Removed checking for
\@@fileswith@pti@ns: Macro	\@unknownversion 100
added 570	1993-11-22 ltlength.dtx LaTeX2e
\@badrequireerror: Macro added 571	\@settodim: Macro added 152
\@fileswithoptions: Added trap	\@settopoint: Macro added 152
for two \LoadClass	\settodepth: Macro added 152
commands. 569	\settoheight: Macro added 152
\@twoloadclasserror: Macro	1993-11-22 ltlogos.dtx LaTeX2e
added 571	\LaTeXe: Macro added 87
\CurrentOption: Name changed	1993-11-23 ltclass.dtx v0.2g
from \@curroption 558	\@useOption: Name changed from
\DeclareOption*: Error checking	\@executeoption 564
added 562	General: Various macros now
\NeedsTeXFormat: Name changed	moved to latex.tex. 557
from \NeedsFormat 566	Warnings and errors now
\ProcessOptions*: restoring	directly coded. 557
\@fileswith@pti@ns added. 563	1993-11-23 ltdefns.dtx LaTeX2e
1993-11-18 ltclass.dtx v0.2d	\@argdef: Macro added 39
\documentstyle: Modified	\@ifundefined: Redefined to
\RequirePackage stuff. 565	remove a trailing \fi 48
\ExecuteOptions: Use	\@newcommand: Macro added 39
\CurrentOption not	\@newenv: Macro interface changed 42
\reserved@a 564	\@xargdef: Macro interface
\NeedsTeXFormat: \fmtname	changed 39
\fmtversion not \@. 566	\@yargd@f: Avoid \@? token . . 40

Macro interface changed	40	1993-12-01 ltoutput.dtx v1.0e	
\newcommand: Macro		\@reinserts: Command added . .	505
reimplemented and extended . .	39	1993-12-03 ltboxes.dtx v0.1a	
\renewcommand: Macro		\@argsbox: macro removed	390
reimplemented and extended . .	41	\@begin@tempboxa: macro added	381
\renewenvironment: Macro		\@end@tempboxa: macro added . .	381
reimplemented and extended . .	42	\@iirsbox: redefined to support	
\two@digits: Macro added	36	\height	390
1993-11-23 ltoutput.dtx v0.1a		\@imakebox: macro modified . . .	381
\paperheight: Register added . .	493	\@irsbox: redefined to support	
\paperwidth: Register added . . .	493	\height	390
1993-11-23 ltoutput.dtx v0.1c		\@isavebox: color support	383
\@enlargepage: Command added	536	extra group	383
\@kludgeins: Insert added	536	\@isavepicbox: extra group	383
\@makecol: Command changed . .	504	\@makebox: default changed from x	
\@specialoutput: Command		to c	381
changed	498	\@makepicbox: macro modified . .	382
\enlargethispage*: Commands		\@savebox: default c not x	383
added	536	\bm@b: macros added	381
1993-11-24 ltfntcmd.dtx v2.1a		\endlrbox: macro added	383
\maybeic@: Use \t@st@ic	297	\fbox: extra group	384
\t@st@ic: Macro added	297	\lrbox: color support	383
1993-11-24 ltfssini.dtx v2.1a		macro added	383
General: Removed \xpt stuff . . .	265	\makebox: modified	380
1993-11-24 ltlogos.dtx LaTeX2e		\mbox: extra group	381
\LaTeX: Macro changed	87	\minipage: Redefined to support	
1993-11-28 ltclass.dtx v0.2h		extra optional arguments . . .	388
\@twoclasseserror: Macro added	571	\newsavebox: Pass the whole of	
General: Assorted commands now		arg 1 to \@ifdefinable	382
in the kernel removed.	557	\parbox: Redefined to support	
Directory syntax checing moved		extra optional arguments . . .	385
to dircheck.dtx	557	\raisebox: redefined to support	
Primitive filenames now		\height	390
terminated by space not		\sbox: color support	383
\relax.	557	extra group	383
\endfilecontents: Don't globally		\set@color: color support	382
allocate a write stream (always		macro added	382
use 15)	571	1993-12-03 ltclass.dtx v0.2i	
1993-11-28 ltfiles.dtx LaTeX2e		\@cls@pkg: Name changed to avoid	
\@missingfileerror: Use filename		clash with output routine. . .	570
parser from dircheck	99	General: \@onlypreamble: Many	
1993-11-29 ltoutput.dtx v1.0b		commands declared.	557
\@makecol: \@makespecialcolbox		Removed obsolete	
added	504	\@documentclass	557
\@makespecialcolbox: Command		1993-12-03 lterror.dtx v1.0b	
added	506	\@latexerr: Set	
1993-11-29 ltplain.dtx LaTeX2e		\@errorcontextlines to -1 . .	65
General: All accents in decimals;		1993-12-03 ltfssini.dtx v2.1a	
suggested by Paul Taylor	29	General: update for LaTeX2e . . .	252
1993-11-30 ltoutput.dtx v1.0c		1993-12-04 ltfiles.dtx v0.9b	
\fl@tracemessage: Commands		\@iinput: Macro reimplemented . .	99
added	538	\@input: Macro reimplemented . .	99
1993-12-01 fontdef.dtx v2.1a		\IfFileExists@: Macro added . .	97
General: Update for LaTeX2e . . .	268	\input: Macro reimplemented . .	98

<code>\InputIfFileExists</code> : Macro added	98	1993-12-06 ltoutput.dtx v1.0f	
1993-12-05 ltfloat.dtx LaTeX2e		<code>\@specialoutput</code> : Unboxing of 255 added to rescue writes	498
<code>\dblfloatplacement</code> : Command changed	462	1993-12-06 ltoutput.dtx v1.0g	
<code>\xfloat</code> : Command changed	457	<code>\@topnewpage</code> : <code>\@floatplacement</code> placement bug fixed	496
1993-12-05 ltoutput.dtx v1.0f		1993-12-07 ltclass.dtx v0.2l	
<code>\@addtobot</code> : Command changed	518	<code>\ProvidesFile</code> : Macro added	562
<code>\@addtocurcol</code> : Command changed	519	1993-12-07 ltclass.dtx v0.2m	
<code>\@addtodblcol</code> : Command changed	530	<code>\@fileswithoptions</code> : Reset <code>\CurrentOption</code>	568
<code>\@addtonextcol</code> : Command changed	526	1993-12-07 ltoutenc.dtx 1.1	
<code>\@addtotoporbot</code> : Command changed	518	General: Protected all special characters with <code>\string</code>	105
<code>\@boxfpsbit</code> : Command added	541	1993-12-07 ltoutenc.dtx v1.1	
<code>\@flcheckspace</code> : Command added	543	General: Made all character numbers decimal.	102
<code>\@flsetnum</code> : Command added	542	Removed a lot of equal signs and the like.	102
<code>\@flsettextmin</code> : Command added	543	1993-12-08 ltboxes.dtx v0.1b	
<code>\@flstop</code> : Commands added	539	<code>\@begin@tempboxa</code> : Extra braces for color support (braces removed from other macros)	381
<code>\@flupdates</code> : Command added	544	<code>\@irsbox</code> : fix typo	390
<code>\@fpsadddefault</code> : Command added	540	<code>\@parboxto</code> : <code>\endgraf</code> added due to extra group in <code>\@begin@tempboxa</code>	386
<code>\@getfpsbit</code> : Command added	541	<code>\lrbox</code> : move <code>\@endpfalse</code> out of the inner group	383
<code>\@opcol</code> : Command changed	504	1993-12-08 ltfncmd.dtx v2.1b	
Hook added	504	General: Macros <code>\rm</code> , <code>\bf</code> and <code>\sf</code> moved to classes.dtx	299
<code>\@outputpage</code> : Command changed	508	1993-12-08 ltlists.dtx LaTeX2e	
<code>\@resethfps</code> : Command added	542	<code>\@item</code> : use <code>\sbox</code> to support colour	376
<code>\@setfloattypecounts</code> : Command added	540	1993-12-08 ltspace.dtx LaTeX2e	
<code>\@setfpsbit</code> : Command added	541	<code>\@bsphack</code> : Command reimplemented	77
<code>\@shipoutsetup</code> : Command added	508	Command reimplemented; late birthday present for Chris	77
<code>\@startcolumn</code> : Command changed	513	<code>\@vbsphack</code> : Command added	80
<code>\@startdblcolumn</code> : Command changed	513	1993-12-09 ltboxes.dtx v0.1c	
<code>\@testfp</code> : Command added	541	<code>\@irsbox</code> : fix another typo	390
<code>\@textfloatsheight</code> : Commands added	540	1993-12-09 ltclass.dtx v0.2n	
<code>\@topnewpage</code> : Commands changed	496	<code>\documentstyle</code> : input 209 compatibility file.	565
<code>\@tryfcolumn</code> : Command changed	515	1993-12-09 ltfiles.dtx v0.9e	
<code>\@writesetup</code> : <code>\@startpagehook</code> added	508	<code>\document</code> : Hook added	90
<code>\output</code> : Command changed	498	1993-12-09 ltmiscen.dtx v0.9e	
1993-12-06 ltclass.dtx v0.2k		<code>\enddocument</code> : Hook added	337
<code>\ExecuteOptions</code> : Preserve <code>\CurrentOption</code>	564	1993-12-10 ltoutenc.dtx v1.2	
		General: Added source code for <code>tlenc.sty</code>	102

1993-12-11 ltfntcmd.dtx v3.0a		1993-12-13 ltclass.dtx v0.2o	
General: Complete reworking of all		General: Removed setting	
text commands, using just one		<code>\errorcontextlines</code> (now in	
creator function	292	<code>latex.tex</code>)	557
italic correction now put in front		<code>\documentstyle</code> : compatibility file	
of penalty before glue	292	now <code>latex209.sty</code>	565
newcommands replaced by defs	292	<code>\usepackage</code> : Fixed error	
newfontswitch command		handling	566
corrected and changed	292	1993-12-13 ltdirchk.dtx v0.2a	
<code>\DeclareTextFontCommand</code> : Macro		General: on the ‘docstrip’ pass, do	
changed	294	not check openin path	10
<code>\emph</code> : Macro changed	295	<code>\IfFileExists</code> : Removed	
<code>\fix@penalty</code> : Macro added	298	interactive prompting for	
<code>\maybe@ic</code> : Macro name changed	297	current directory syntax	10
<code>\maybe@ic@</code> : Macro and name		<code>\strip@prefix</code> : modified, name	
changed	297	changed from <code>\stripmeaning</code>	5
<code>\sw@slant</code> : Macro changed	298	1993-12-13 ltlists.dtx latex2e	
<code>\textup</code> : Macros changed	295	<code>\trivlist</code> : Initialised	
1993-12-11 ltmath.dtx v0.9g		<code>\@itemlabel</code>	373
General: Added a group around		1993-12-13 ltmiscen.dtx v0.9h	
the first argument of <code>\frac</code> to		<code>\@noligs</code> : Readded <code>\@noligs</code>	350
prevent changes (for example		<code>\@verbatim</code> : Readded <code>\@noligs</code>	347
font changes) from modifying		Removed optional argument of	
the contents of the second		<code>\item</code>	346
argument.	359	<code>center</code> : Removed optional	
1993-12-11 ltoutenc.dtx v1.2a		argument of <code>\item</code>	345
General: Corrected for <code>tlenc</code> ,		<code>flushleft</code> : Removed optional	
<code>math</code>	102	argument of <code>\item</code>	345
1993-12-11 ltsect.dtx LaTeX2e		<code>flushright</code> : Removed optional	
<code>\author</code> : Added default	441	argument of <code>\item</code>	345
<code>\title</code> : Added default	441	1993-12-13 ltoutenc.dtx v1.2b	
1993-12-11 ltxref.dtx LaTeX2e		General: Corrected file name in	
<code>\@setref</code> : Macro added	334	driver code.	102
<code>\pageref</code> : Macro reimplemented	334	1993-12-13 lttab.dtx latex2e	
<code>\ref</code> : Macro reimplemented	334	<code>\tabbing</code> : Removed optional	
1993-12-12 ltoutput.dtx v1.0h		argument of <code>\item</code>	398
<code>\@cflb</code> : boxmaxdepth setting		1993-12-14 ltoutput.dtx v1.0i	
moved	512	General: Section added to declare	
defs changed to <code>lets</code>	512	all parameters	549
<code>\@cflt</code> : name changed	512	1993-12-15 ltboxes.dtx v0.1d	
<code>\@docclearpage</code> : defs changed to		<code>\@iminipage</code> : Changed default	
<code>lets</code>	503	from ‘c’ to ‘s’	388
<code>\@makecol</code> : defs changed to <code>lets</code>	505	<code>\@iparbox</code> : Changed default from	
<code>\@resetfhps</code> : Warnings added:		‘c’ to ‘s’	386
minimal	542	<code>\minipage</code> : Changed default from	
<code>\@startdblcolumn</code> : defs changed		‘c’ to ‘s’	388
to <code>lets</code>	514	extra space removed.	388
<code>\@topnewpage</code> : braces removed	496	<code>\parbox</code> : Changed default from ‘c’	
<code>\@tryfcolumn</code> : defs changed to		to ‘s’	385
<code>lets</code>	515	1993-12-15 ltclass.dtx v0.2p	
<code>\fl@tracemessage</code> : Commands		General: Removed extra ‘.’s from	
changed	538	<code>\@warnings</code>	557
		1993-12-16 ltlogos.dtx LaTeX2e	
		<code>\LaTeXe</code> : Extended logo by DPC	87

1993-12-16 ltmath.dtx v0.9i		\@writesetup: —and then	
\@eqnrcr: use \refstepcounter		removed	508
instead of shortcut	360	\fl@tracemessage: tracefloatvals	
General: use \refstepcounter		made a document command	538
instead of shortcut	359	1993-12-17 ltpage.dtx LaTeX2e	
1993-12-16 ltmiscen.dtx v0.9i		\mark: Removed init \mark at	
General: \literal added	350	begin document, since it	
1993-12-16 ltpage.dtx LaTeX2e		doesn't work.	480
\mark: Init \mark at begin		\rightmark: Stopgap solution to	
document	480	mark \leftmark and	
1993-12-16 ltspace.dtx LaTeX2e		\rightmark work without	
\@bsphack: Corrected optimisation		initializing mark until the	
:-)	77	problem is solved.	480
1993-12-16 lttab.dtx latex2e		1993-12-18 ltoutenc.dtx 1.3b	
\@xhline: Measure from middle of		General: Fixed typos with	
vertical rules	412	\ProvidesPackage lines.	
1993-12-17 ltclass.dtx v0.2q		Added the \NeedsTeXFormat	
\@documentclasshook: Macro		line. Added the last argument	
added	557	to \DeclareEncoding. Moved	
\@fileswithoptions: Add		the use of the encodings to	
\@compatibility hook	567	after their declaration.	105
\documentstyle: Match Alan's		Replaced the missing last	
new code.	565	argument to	
1993-12-17 ltoutenc.dtx 1.3		\DeclareFontEncoding.	118, 120
General: Added this section	106	1993-12-18 ltoutenc.dtx 1.3c	
Removed all the hackery for use		General: Rewrote for the new	
in \DeclareFontEncoding, and		syntax of	
redid everything using		\EncodingSpecific.	118, 120
\DeclareTextFoo.	118, 120	Split \EncodingSpecificAccent	
Removed the catcode hackery,		up into \EncodingSpecific	
since the file is only read as a		and \DeclareAccent.	106
package in the preamble, and		1993-12-18 ltoutenc.dtx v1.3a	
removed all the messages on		General: Replaced OT3 by XXX	102
the screen, which just confuse		1993-12-18 ltoutenc.dtx v1.3b	
users. Replaced them by the		General: Corrected typos.	102
appropriate \ProvidesPackage		Replaced the missing last	
commands. Added XXXenc. . . .	105	argument to	
1993-12-17 ltoutenc.dtx v1.3		\DeclareFontEncoding.	102
General: Added		1993-12-18 ltoutenc.dtx v1.3c	
\EncodingSpecificAccent,		General: A new syntax, separating	
\EncodingSpecificAccent-		accent-definitions from	
edLetter and		encoding-specific definitions,	
\EncodingSpecificCommand. . . .	102	and allowing encoding-specific	
Made Rokicki's encoding a		\chardef, \let, etc.	102
proper encoding scheme rather		Rewrote for the new syntax of	
than a variant of OT1.	102	\EncodingSpecific.	102
1993-12-17 ltoutput.dtx v1.0j		1993-12-18 ltoutenc.dtx v1.3d	
\@opcol: Hook removed	504	General: Some T1 stuff had drifted	
\@specialoutput: Page room test		into the OT1 file.	102
added	499	1993-12-18 ltpage.dtx LaTeX2e	
\@topnewpage: check for vsize too		\sloppy: Added	
small added	496	\emergencystretch	480
Page room test added	498		

- 1993-12-19 ltclass.dtx v0.2r
`\endfilecontents`: Different message when ignoring a file 571
- 1993-12-19 ltfntcmd.dtx v3.0b
 General: `\@pdef` command added 292
 Added by ASAJ. 300
 Made `\@newfontswitch` produce an error if command already exists, and added
`\@renewfontswitch`, ASAJ 292
 Other tidying 292
 Some more tidying done 292
 Untidying added, so this is now a TEMPORARY version. 292
 Wording changes by CAR. 300
- `\DeclareOldFontCommand`:
 Corrected and tidied 299
- `\DeclareTextFontCommand`:
 Corrected and tidied 294
- 1993-12-19 ltspace.dtx LaTeX2e
`\@bsphack`: There seem to be problems with selfmade birthday presents 78
- 1993-12-20 ltdefns.dtx LaTeX2e
`\@reargdef`: Kept old version of `\@reargdef`, for array.sty 41
- 1993-12-20 ltfiles.dtx v0.9m
`\@obsoletefile`: Added this command, removed
`@oldfilewarning` 100
- 1994-01-05 fontdef.dtx v2.1d
 General: Removed nf prefix from file names. 270
- 1994-01-13 ltmath.dtx v0.9o
`\@eqncr`: correcting 0.9i 360
 General: correcting 0.9i 359
- 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 92
- 1994-01-17 ltclass.dtx v0.2s
`\@fileswithoptions`: Modify to reduce parameter stack usage 567, 569
 General: Added many more
`\@onlypreamble` commands 557
 Wrapped long lines to column 72 557
- 1994-01-17 ltfiles.dtx LaTeX2e
`\listfiles`: New Version, adds ‘.tex’ if needed, and lines up columns 100
- 1994-01-17 ltssbas.dtx v2.1a
 General: New math font
 setup 153, 176
`\curr@math@size`: New math font
 setup 163, 186
`\everydisplay`: New math font
 setup 162, 185
`\everymath`: New math font
 setup 162, 185
`\frozen@everydisplay`: New math font
 font setup 163, 186
`\frozen@everymath`: New math font
 font setup 162, 185
`\math@version`: New math font
 setup 162, 185
- 1994-01-17 ltssini.dtx v2.1e
`\not@math@alphabet`: Message changed 264
- 1994-01-17 ltssstrc.dtx v2.3a
 General: New math font setup 199
`\check@mathfonts`: New math font
 setup 208
`\glb@currsz`: New math font
 setup 205
`\restglb@settings`: New math font
 font setup 208
- 1994-01-18 ltbibl.dtx LaTeX2e
`\bibliography`: Use `\@input@` so include files are listed. 476
- 1994-01-18 ltclass.dtx v0.2t
`\@ifclassloaded`: Fix typo
`\@pkgetension` 558
- 1994-01-18 ltfiles.dtx v0.9p
`\@iffileonpath`: Macro added 97
`\@input`: do not use a different definition for `\input@path` 99
`\@input@`: Macro added 99
`\IfFileExists@`: New Definition 97
`\include`: Use `\@input@` so include files are listed. 94
`\InputIfFileExists`: New Definition 98
- 1994-01-18 ltssini.dtx v2.1f
`\not@math@alphabet`: Message corrected 264
- 1994-01-18 ltmiscen.dtx v0.9p
`\@verbatim`: Add
`\global\@inlabelfalse` 346
 Only add `\penalty` if in hmode 346
- 1994-01-19 fontdef.dtx v2.1e
 General: Added missing setting for symbols in bold version. 274

1994-01-19 ltdirchk.dtx v0.2e	1994-01-27 ltfnctcmd.dtx v3.1a
\IfFileExists: name changed	\nocorrlist: Only ., used as
from \test 9	default for cm fonts 298
\input@path: No longer check that	1994-01-29 ltclass.dtx v0.2v
an empty group is in the path 11	\@unprocessedoptions: Macro
\strip@prefix: name changed	added. 571
from \strip@meaning, to	\@fileswithoptions: All options
match NFSS. 5	raise error if no
1994-01-19 ltmath.dtx v1.0n classes	\ProcessOptions appears . . 569
\mathindent: Deferred setting of	1994-01-31 ltclass.dtx v0.2w
\mathindent 361	\g@addto@macro: Use toks register
1994-01-20 ltdirchk.dtx v0.2f	to avoid ‘hash’ problems . . . 570
General: \copytexsys and the	1994-01-31 ltfiles.dtx v0.9t
texsys.new file removed 9	\document: set \normalsize or
Modify all of ltxcheck 14	\normalsize if necessary . . . 91
\IfFileExists: \copytexsys	1994-01-31 ltfnctcmd.dtx v3.1b
removed 10	General: \normalsize no longer
1994-01-21 ltclass.dtx v0.2u	defined 292
\documentstyle: compatibility file	1994-02-01 ltpage.dtx LaTeX2e
now latex209.def. 565	\pagestyle: (DPC) Modify to get
1994-01-21 ltdirchk.dtx v0.2g	nicer error message 478
General: Improve documentation,	\thispagestyle: (DPC) Modify to
reorganise docstrip module . . . 1	get nicer error message 479
\filename@parse: Minor changes,	1994-02-02 ltclass.dtx v0.2x
and add Mac version (:) 11	\@fileswithoptions: Only run
\today: Name changed from	the hook and options check if
\stamp, to save memory 9	the file was loaded. 569
1994-01-21 ltfloat.dtx LaTeX2e	1994-02-03 ltoutput.dtx v1.0k
\@xfloat: Added missing percent	\@makespecialcolbox: correct
characters. 457	mistakes in the
1994-01-21 ltmiscen.dtx v0.9s	documentation 507
\verbatim@font: Removed	1994-02-07 ltclass.dtx v0.2y
unnecessary category code	\@fileswithoptions: Run
hackery. 347	\@compatibility on the first
1994-01-24 ltdirchk.dtx v0.2h	class to start (not the first to
\IfFileExists: Stop testing once	finish) 567
texsys.aux has been found . . . 10	\@ifclasswith: Add extra ,s so
1994-01-24 ltpage.dtx LaTeX2e	‘two’ is not matched with
\pagestyle: (DPC) Complain if	‘twocolumn’ 560
pagestyle is undefined. 478	\ProcessOptions*: Add extra ,s so
1994-01-25 ltdirchk.dtx v0.2i	‘two’ is not matched with
General: Protect against looping	‘twocolumn’ 563
on \@@input and \@@end. 2	1994-02-07 ltfsbas.dtx v2.1c
1994-01-25 ltfsbas.dtx v2.1b	\DeclareFontEncoding: revert
\math@version: Corrections for	catcode settings earlier . . 155, 178
math setup 162, 185	\DeclareFontShape: revert
1994-01-25 ltmath.dtx LaTeX2e	catcode settings earlier . . 154, 177
\bordermatrix: Removed	1994-02-08 ltoutput.dtx v1.0k
\p@renwd. 355	\@makespecialcolbox:
1994-01-26 ltfsstrc.dtx v2.3c	boxmaxdepth setting added . . 506
\check@mathfonts: Correct trace	boxmaxdepth setting removed 506
info placement 208	General: Documentation and tasks
\restglb@settings: Correct trace	tidied. 481
info placement 208	

1994-02-10 ltclass.dtx v0.2z	1994-03-04 ltvers.dtx v1.0a
\@documentclasshook: Changed the name from	General: Initial version, split from latex.dtx 33
\@compatibility to	1994-03-07 ltboxes.dtx v0.1a
\@documentclasshook, and	\@mpfootnotetext: Extra group for colour 389
added the check for whether	1994-03-07 ltboxes.dtx v1.0a
\@normalsize has been	General: Unify format with other Kernel files 380
defined. ASAJ. 557	1994-03-07 ltdefs.dtx v1.0a
\@fileswithoptions: Renamed	\@italiccorr: Macro added . . . 36
\@compatibility to	1994-03-07 ltfiles.dtx v1.0a
\@documentclasshook. ASAJ. 567	General: Initial version, split from latex.dtx 88
1994-02-10 ltffsbas.dtx v2.1d	Long lines wrapped to 72 columns 88
\addto@hook: Made \addto@hook long. 175, 198	1994-03-07 ltfinal.dtx v0.1a
1994-02-10 ltffscmp.dtx v2.1d	General: Add code from the old dump.dtx 625
\scan@fontshape: scan away stuff after pt 223	Initial version, split from latex.dtx 613
1994-02-22 ltffsini.dtx v2.1g	move code here from lhyphen.dtx 618
General: Correct error message . . 266	Remove oldcomments environment 613
1994-02-24 ltffsbas.dtx v2.1e	use \InputIfFileExists not \IfFileExists 618
\DeclareFontShape: Separate restoration of catcodes for fd cmds 154, 177	1994-03-07 ltfloat.dtx v1.0a
\define@newfont: Separate restoration of catcodes for fd cmds 164, 187	\@endfloatbox: (DPC) Extra group for colour 462
\nfss@catcodes: Separate restoration of catcodes for fd cmds 165, 188	\@footnotetext: (DPC) Extra group for colour 470
1994-02-25 ltdirchk.dtx v0.2j	\@xfloat: (DPC) Extra group for colour 458
General: Remove need for drv file . 1	1994-03-07 lthyphen.dtx v0.1c
1994-03-01 ltdirchk.dtx v0.2k	General: move the 2kernel code to ltfinal.dtx 586
General: Add unstripped module, so that dircheck.dtx may be used with initex 1	1994-03-07 ltlength.dtx v1.0a
1994-03-02 ltboxes.dtx v0.1e	\@settodim: (DPC) Extra group for colour 152
General: Add 2kernel module . . 380	1994-03-07 ltlists.dtx v1.0a
Remove need for drv file 380	General: Initial version, split from latex.dtx 365
1994-03-02 ltclass.dtx v0.3a	Long lines wrapped to 72 columns 365
General: Remove need for driver file 557	1994-03-07 ltpage.dtx v1.0a
1994-03-03 ltboxes.dtx v0.1f	General: Initial version, split from ltherest.dtx 478
\@irsbox: Replaced a missing \else 390	1994-03-07 ltpictur.dtx v0.1a
1994-03-04 ltfloat.dtx v1.0a	General: Initial version, split from latex.dtx 414
General: Initial version, split from latex.dtx 453	Long lines wrapped to 72 columns 414
1994-03-04 ltsect.dtx v1.0a	
General: Initial version, split from latex.dtx 441	
1994-03-04 lttab.dtx v1.0a	
General: Initial version, split from latex.dtx 392	

1994-03-07 ltsect.dtx v1.0a	1994-03-12 ltplain.dtx v1.0b
\@hangfrom: (DPC) Extra groups for colour 448	General: Name changed from lplain. The end of an era 15
1994-03-07 lttab.dtx v1.0a	1994-03-12 ltplain.dtx v1.0e
General: Long lines wrapped to 72 columns 392	General: Replaced remaining width, height, depth by L ^A T _E X macro names to save tokens. . . 15
1994-03-08 ltclass.dtx v0.3b	1994-03-13 ltcntrl.dtx v1.0c
General: Modify driver code into ‘new style’ 557	\@tfor: (DPC) Add \@tfor so a single group is correctly treated. 59
1994-03-08 ltdirchk.dtx v1.0a	1994-03-13 ltfiles.dtx LaTeX2e
General: Reorganise driver module into ‘new style’ 1	\@addtofilelist: Macro added . . 100
1994-03-08 ltplain.dtx v1.0a	\listfiles: Reset \@addtofilelist at begin document 101
General: Remove need for a driver file. 15	1994-03-13 ltfiles.dtx v0.3b
1994-03-10 ltfsbas.dtx v2.2f	\InputIfFileExists: Use new cmd \@addtofilelist 98
\math@egroup: Changed \begin group/\end group to \bgroup/\egroup. 172, 195	1994-03-13 ltfsbas.dtx v2.1g
1994-03-11 ltssdcl.dtx v2.1b	General: add 2kernel module to omit repeated code 153, 176
\DeclareSymbolFontAlphabet@: Added check against use of alphabet switch outside of math mode. 251	1994-03-13 ltssdcl.dtx v2.1c
\SetMathAlphabet@: Changed parameter template in temporary macro to catch check add below. 239	General: add 2kernel module to omit repeated code 226
1994-03-12 ltclass.dtx v0.3c	1994-03-14 ltboxes.dtx v1.0b
\@fileswithoptions: Do not use \@pr@videpackage to avoid typeout 569	\@isavebox: Use \color@setgroup 383
General: Change name from docclass to ltclass 557	\@isavepicbox: Use \color@setgroup 383
\ProvidesFile: Add \wlog . . . 562	\color@begin group: macro added for colour support 382
\ProvidesPackage: Add \wlog . . 561	\color@end group: macro added for colour support 382
use \@gtempa 561	\lrbox: Use \color@setgroup . . 383
1994-03-12 ltdefns.dtx v1.0b	\sbox: Use \color@setgroup . . . 383
\@reargdef: New defn, in terms of \@yargdef 41	1994-03-14 ltfloat.dtx 1.0c
\@yargdef: Name changed from \XXX@argdef 40	\@xympar: (DPC) Use \color@begin group 466
1994-03-12 ltdirchk.dtx v1.0b	1994-03-14 ltfloat.dtx v1.0c
General: Change name from dircheck.dtx 1	\@endfloatbox: (DPC) Use \color@end group 462
Minor edits to the typeouts in ltxcheck 1	\@footnotetext: (DPC) Use \color@begin group, add \endgraf 470
1994-03-12 ltfloat.dtx v1.0b	\@savemarbox: (DPC) Use \color@begin group 465
\@savemarbox: (DPC) Extra group for colour 465	\@xfloat: (DPC) Use \color@begin group 458
\@xympar: (DPC) Extra bgroup for colour 466	1994-03-15 ltfiles.dtx LaTeX2e
	\@missingfileerror: Quit on x or X just like a real error 99

1994-03-15 ltfnctcmd.dtx v3.2a	1994-03-28 ltfiles.dtx v1.0d
General: Adapted to mass	\document: (DPC) Use
formatting 292	\normalsize not
Changed \ / to \@@italiccorr 292	\@normalsize 91
Removed \@renewfontswitch . 292	(DPC) remove \@normalsize
Removed defs of short-forms and	check 91
all sizes except \normalize . 292	1994-03-28 ltfloat.dtx v1.0b
1994-03-15 ltoutput.dtx v1.0l	\@caption: Use \normalsize not
\@addtocurcol: Changed	\@normalsize 456
\addvspace to \vskip . . 521, 525	General: Split further from
\@combinedblfloats: Removed	ltherest.dtx 453
boxmaxdepth setting. 513	1994-03-28 ltlists.dtx v1.0b
\@makecol: \maxdepth changed to	General: Improve documentation 364
\@maxdepth 504	1994-03-28 ltmiscen.dtx v1.0c
Removed boxmaxdepth setting. 505	General: Improve Documentation 336
\@makespecialcolbox: Removed	1994-03-28 ltplain.dtx v1.0c
boxmaxdepth setting. 506	\newlanguage: Remove some
\@topnewpage: Corrected and	\outer declarations. 17
amended warning message . . 497	1994-03-28 ltsect.dtx v1.0b
Warning added: it should be	General: Split further from
improved 498	ltherest.dtx 441
General: Added some warnings	1994-03-28 lttab.dtx v1.0b
when page gets full of top	General: Improve documentation 392
floats. 481	1994-03-28 ltthm.dtx v1.0a
Driver added and further	General: Initial version, split from
tidying. 481	latex.dtx 437
Removed duplicated code and	1994-03-29 ltcunts.dtx v1.0c
corrected docstrip options. . . 481	General: Create file from parts of
Some boxmaxdepth settings	ltmiscen and ltherest. 145
removed. 481	1994-03-29 ltlength.dtx v1.0c
1994-03-16 ltclass.dtx v0.3f	General: Create file ltcntlen from
General: Add pkgindoc package . 585	parts of ltmiscen and ltherest. 152
1994-03-16 ltfiles.dtx LaTeX2e	1994-03-29 ltmiscen.dtx v1.0d
\listfiles: Move this code	General: Remove counter macros
directly into \document 101	to ltcntlen 336
1994-03-16 ltfiles.dtx v1.0c	1994-03-29 ltpageno.dtx v1.0c
\document: (DPC) directly add file	General: Create file ltcntlen from
list settings 91	parts of ltmiscen and ltherest. 331
1994-03-16 ltmiscen.dtx v1.0b	1994-03-29 ltxref.dtx v1.0c
\@verbatim: Remove	General: Create file ltcntlen from
\global\@inlabelfalse	parts of ltmiscen and ltherest. 332
again. 346	1994-03-31 ltbibl.dtx v1.0a
1994-03-28 ltalloc.dtx v1.0d	General: Initial version of
General: Redefinition of ‘new’	ltxbib.dtx, split from
allocations removed. 54	ltherest.dtx 474
1994-03-28 ltdirchk.dtx v1.0d	1994-03-31 ltxglo.dtx v1.0a
General: Improve documentation . 1	General: Initial version of
1994-03-28 lterror.dtx v1.0d	ltxbib.dtx, split from
\@invalidchar: (DPC) Comment	ltherest.dtx 472
out (use catcode15 instead) . . 67	1994-04-09 ltcunts.dtx v1.0d
General: Remove test for	\@newctr: \@nocnterr now has
\inputlineno undefined. 64	counter name argument 146
	\@addtocounter: \@nocnterr now
	has counter name argument . 146

<code>\setcounter</code> : <code>\nocnterr</code> now has counter name argument	146	(without arg) kept	65
<code>\stepcounter</code> : Use <code>\addtocounter</code> to have name checked	146	1994-04-17 ltthm.dtx v1.0c	
1994-04-09 ltthm.dtx v1.0b		<code>\@othm</code> : Use new std counter error message (FMi)	439
<code>\@othm</code> : Use standard counter error message (FMi)	439	1994-04-18 ltfinal.dtx v0.1b	
1994-04-11 ltclass.dtx v0.3g		General: Initialise <code>\textheight</code> , <code>\textwidth</code> and page style . .	615
<code>\endfilecontents</code> : Add star form, dont write <code>\endinput</code> at the end of the file.	571	1994-04-18 ltfloat.dtx v1.0d	
<code>\ProvidesFile</code> : Protect against weird catcodes.	562	<code>\@footnotetext</code> : (DPC) Remove Colour support	470
1994-04-11 ltffsbas.dtx v2.1h		<code>\@savemarbox</code> : (DPC) Remove Colour support	465
General: Added		1994-04-18 ltffsbas.dtx v2.1i	
<code>\defaultscriptratio</code> and <code>\defaultscriptscriptratio</code> . ASAJ.	153, 176	General: Macro	
<code>\defaultscriptratio</code> : Macro added	173, 196	<code>\no@alphabet@help</code> removed again	153, 176
<code>\defaultscriptscriptratio</code> : Macro added	173, 196	<code>\calculate@math@sizes</code> : Changed message to log only	173, 196
1994-04-12 ltboxes.dtx v1.0c		<code>\no@alphabet@error</code> : Use std LaTeX error macro	153, 176
General: Remove <code>\@acci</code> , now defined in ltplain.dtx	386	1994-04-18 ltffsdcl.dtx ???	
Remove <code>\@dischyph</code> , now defined in ltinit.dtx	386	<code>\DeclareMathAlphabet</code> : Pass correct arg (2 not 3)	237
1994-04-12 ltdefs.dtx v1.0g		1994-04-18 ltffsdcl.dtx v2.1d	
<code>\@dischyph</code> : Define <code>\@dischyph</code> , was previously in ltboxes.dtx .	51	General: Removed surplus <code>\no@alphabet@error</code> (see fam.dtx)	226
1994-04-12 ltplain.dtx v1.0d		1994-04-18 ltfsstrc.dtx v2.3d	
General: Define <code>\@acci</code>	29	General: Changed to new error/warning scheme	199
1994-04-12 ltvers.dtx v1.0b		<code>\font@submax</code> : Changed dimen to macro	216
General: Have version info generated automatically. . . .	33	<code>\fontsubfuzz</code> : Changed dimen to macro	216
1994-04-14 ltftncmd.dtx v3.2b		<code>\subst@size</code> : <code>\font@submax</code> and <code>\fontsubfuzz</code> now macros . .	217
General: Macros renamed to non-private forms, JB	292	1994-04-19 ltpage.dtx v1.0b	
<code>\DeclareOldFontCommand</code> : Renamed from		General: Improve documentation	478
<code>\@newfontswitch</code>	299	1994-04-20 ltftncmd.dtx v3.3a	
1994-04-15 ltboxes.dtx v1.0d		General: Documentation up-dated	292
<code>\@isavebox</code> : Added missing procent character.	383	New implementation of <code>\nocorr</code>	292
1994-04-17 ltcounts.dtx v1.0e		<code>\check@nocorr@</code> : Macros added .	296
<code>\@newctr</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	146	<code>\maybe@ic@</code> : <code>\nocorr</code> etc removed from list of tokens to check, leaving only punctuation characters	297
<code>\addtocounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	146	1994-04-20 ltmiscen.dtx v1.0e	
<code>\setcounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	146	<code>\enddocument</code> : Changed logic for producing warning messages	338
1994-04-17 ltterror.dtx v1.0h		1994-04-21 ltboxes.dtx v1.0e	
<code>\@nocounterr</code> : New name for error message, old error message		<code>\@iiiminipage</code> : Extra <code>\bgroup</code> for colour	388

<code>\@mpfootnotetext</code> : Extra	1994-04-28 lterror.dtx v1.1c
<code>\endgraf</code> for colour 389	<code>\@inmatherr</code> : Replaced <code>\noexpand</code> with <code>\protect</code> 67
<code>\endminipage</code> : Extra <code>\egroup</code> for colour 388	1994-04-28 ltfsdcl.dtx v2.1e
1994-04-21 ltfinal.dtx v0.1c	General: Removed all <code>\uppercase</code> in hex num parsing macros . 226
General: Added comments, set the catcodes of 128–255. 613	1994-04-28 ltlists.dtx v1.0c
1994-04-22 ltfsini.dtx v2.1g	<code>\item</code> : Replaced <code>\@ltxnomath</code> by <code>\@inmatherr</code> 374
<code>\not@math@alphabet</code> : Message changed again 264	1994-04-28 ltpictur.dtx v0.1c
1994-04-23 ltfinal.dtx v0.1d	General: bezier curves added . . . 434
General: Check that <code>\font@submax</code> is still zero 613	<code>\multiput</code> : (DPC) Ignore spaces between <code>)</code> (. 416
1994-04-24 ltoutput.dtx v1.0m	(DPC) Macro added 416
<code>\@resetthfps</code> : Number 2 changed to <code>\tw@</code> 542	<code>\picture</code> : (DPC) Ignore spaces before <code>(</code> 415
Warning changed 542	1994-04-28 ltplain.dtx v1.0g
<code>\@specialoutput</code> : Message changed to give more info and ‘top’ removed 499	General: Turn off overfull box tracing in log 25
<code>\@topnewpage</code> : Message changed to give more info 498	1994-04-29 ltclass.dtx v1.0a
Warning message removed as it will be generated later 497	General: Change version number to 1 (no other change) 557
General: Changed <code>\@normalsize</code> to <code>\normalsize</code> 481	1994-04-29 ltmiscen.dtx v1.0f
Corrected unverbbed commands in documentation. 481	<code>\@verbatim</code> : <code>\leavevmode</code> added 346
Removed some long lines and other aesthetic changes. . . . 481	Change to <code>\everypar</code> added . . 347
Warning messages changed/corrected. 481	1994-04-29 ltoutenc.dtx 1.4a
1994-04-24 ltpictur.dtx v0.1b	General: Removed
General: Removed surplus spaces after <code>\hbox to</code> in several cases 414	<code>\EncodingSpecific</code> . Renamed all the commands. Added <code>\DeclareTextGlyph</code> and <code>\UndeclareTextCommand</code> . . . 106
1994-04-25 ltclass.dtx v0.3h	Removed Rokicki’s OT1 variant encoding. Moved the driver to the top. 105
General: Removed spurious extra ‘.’s at the end of error messages 557	1994-04-30 ltfontcmd.dtx v3.3b
1994-04-25 ltfloat.dtx v1.0e	General: Documentation up-dated and tidied 292
<code>\@largefloatcheck</code> : Changed warning message to give more info 462	Prefix <code>frag@</code> changed to <code>frag</code> in <code>\@protecteddef</code> 292
Command added 462	Title changed 292
General: Changed warning messages 453	Warning changed to info message in <code>\@protecteddef</code> . 292
Removed obsolete tracing code 453	1994-04-30 ltoutput.dtx v1.0n
1994-04-27 ltfsstrc.dtx v2.3e	<code>\@activechar@info</code> : <code>\@activechar@warning</code> changed to <code>\@activechar@info</code> 507
General: Corrected item that was forgotten in last change. . . . 199	<code>\@combinedblfloats</code> : Removed rule in <code>topnewpage</code> case 513
1994-04-28 lterror.dtx v1.0j	<code>\@emptycol</code> : Empty column action added: <code>\@emptycol</code> 496
<code>\@inmatherr</code> : Macro added 67	<code>\@flsetnum</code> : Rogue space removed 542

<code>\@specialoutput</code> : Cut-off point changed to <code>2\baselineskip</code> .	499	1994-05-02 <code>ltdefs.dtx</code> v1.1f	<code>\renewcommand</code> : Removed surplus <code>\space</code> in error	41
Empty column action added: <code>\@emptycol</code>	499		<code>\renewenvironment</code> : Removed surplus <code>\space</code> in error	42
Extra empty column added for twocolumn case	499	1994-05-02 <code>ltfiles.dtx</code> v1.0f	<code>\@iffilenamepath</code> : <code>\@break@loop</code> renamed to <code>\@break@tfor</code> . . .	97
Extra empty column added for twocolumn case (wrong, see below)	499		<code>\@obsoletefile</code> : Make <code>\@onlypreamble</code>	100
<code>\@topnewpage</code> : Added setting of <code>\col@number</code>	496	1994-05-02 <code>ltfinal.dtx</code> v0.1e	General: Added setting the ‘letter’ catcodes.	623
Cut-off point changed to <code>3\baselineskip</code>	498		Added setting the ‘other’ catcodes.	623
Empty column action added: <code>\@emptycol</code>	498		Added setting the special catcodes.	622
Message changed for Frank . . .	498		Made slot 127 illegal	623
General: <code>\@activechar@warning</code> changed to an info message. .	481		Set all the catcodes	613
Added <code>\col@number</code>	481	1994-05-02 <code>ltfinal.dtx</code> v0.1f		
Documentation tidied.	481		General: Set the catcode of control-J.	622
Empty column action added. .	481	1994-05-02 <code>ltmiscen.dtx</code> v1.0g	General: Changed 91 to 1991 and moved some bits	336
Fixed bug from <code>\dblfigrule</code> with <code>\@topnewpage</code>	481	1994-05-02 <code>ltoutput.dtx</code> v1.0o	<code>\@resetfps</code> : Code shortened . . .	542
Full of floats action improved. .	481		General: Code of <code>\@resetfps</code> shortened.	481
<code>\col@number</code> : Added <code>\col@number</code>	493	1994-05-03 <code>ltbibl.dtx</code> v1.0b	<code>\nocite</code> : Make <code>\nocite</code> issue a warning for an undefined citation key.	476
<code>\onecolumn</code> : Added setting of <code>\col@number</code>	495	1994-05-03 <code>ltfinal.dtx</code> v0.1f	General: Set the catcode of control-J to be ‘other’, for use in messages.	613
1994-05-01 <code>lterror.dtx</code> v1.0k		1994-05-03 <code>ltfloat.dtx</code> v1.0f	General: (CAR) Added <code>\@largefloatcheck</code>	453
<code>\@latexerr</code> : (CAR) Added draft <code>\@latexinfo</code>	65		Removed unnecessary braces from arguments of <code>\@ifnextchar</code>	453
1994-05-01 <code>ltoutenc.dtx</code> 1.4a			<code>\end@dblfloat</code> : <code>\@largefloatcheck</code> added . .	461
General: Added the <code>\a</code> command. .	114		<code>\end@float</code> : (CAR) Added <code>\@largefloatcheck</code>	460
Added the <code>\SaveAtCatcode</code> and <code>\RestoreAtCatcode</code> commands.	118	1994-05-03 <code>ltfssdcl.dtx</code> v2.1f	General: Renamed <code>\@DeclareMathDelimiter</code> to <code>\@DeclareMathDelimiter</code> . .	226
Removed the uc/lc table settings, since the T1 uc/lc table is now the default. . . .	125		1994-05-03 <code>ltlists.dtx</code> v1.0d	
Rewrote for the new syntax.	118, 120		<code>\@item</code> : <code>\hskip</code> changed to <code>\kern</code> .	375
1994-05-01 <code>ltoutenc.dtx</code> v1.4a				
General: Removed Rokicki’s encoding.	102			
Renamed the commands, removed the <code>\EncodingSpecific</code> command. Turned all slots into decimal. Added <code>\a</code>	102			
1994-05-02 <code>ltcntrl.dtx</code> v1.0l				
<code>\@break@tfor</code> : Macro added (from <code>ltfiles.dtx</code>)	59			

<p><code>\item</code>: Removed superfluous braces 374</p> <p>1994-05-03 ltmiscen.dtx v1.0h</p> <p><code>\@centercr</code>: <code>\@badcrerr</code> replaced by <code>\@nolnerr</code> 344</p> <p>1994-05-03 lttab.dtx v1.0d</p> <p><code>\@endpbox</code>: Use <code>\@finalstrut</code> based on depth of <code>\@arstrutbox</code> 413</p> <p>1994-05-04 ltclass.dtx v1.0b</p> <p><code>\NeedsTeXFormat</code>: Changed wording of the warning 566</p> <p>1994-05-04 lterror.dtx v1.0m</p> <p><code>\@badcrerr</code>: Error message removed 67</p> <p>1994-05-05 ltbibl.dtx v1.0c</p> <p><code>\@citex</code>: Set switch for warning and end of run. 475</p> <p><code>\nocite</code>: Do not write page number in <code>\nocite</code> warning message. 476</p> <p>Set switch for warning and end of run. 476</p> <p>1994-05-05 ltfinal.dtx v0.1g</p> <p>General: Added empty <code>errhelp</code>. . . 613</p> <p><code>\errhelp</code>: Set error help empty. . 626</p> <p>1994-05-05 ltfntcmd.dtx v3.3c</p> <p><code>\@math@egroup</code>: Corrected <code>\@fontswitch</code> and added saved versions 299</p> <p>General: Corrected <code>\@fontswitch</code> 292</p> <p>1994-05-05 ltmiscen.dtx v1.0i</p> <p>General: Removed braces from <code>ifnextchar</code> and <code>ifstar</code> arguments 336</p> <p>1994-05-07 lttab.dtx v1.0c</p> <p><code>\@maxtab</code>: Changed <code>\@firsttab</code> to <code>\chardef</code> 396</p> <p>Changed <code>\@maxtab</code> to <code>\chardef</code> 396</p> <p>General: Removed definition of <code>\+</code> Removed surplus braces from <code>\@ifnextchar</code> constructs . . . 392</p> <p>1994-05-08 ltfntcmd.dtx v3.3d</p> <p>General: Removed <code>\@undefinedfonterror</code> 292</p> <p><code>\normalsize</code>: Removed <code>\@undefinedfonterror</code> 300</p> <p>1994-05-09 ltfntcmd.dtx v3.3f</p> <p>General: Replaced all <code>\next</code> by <code>\@let@token</code> and undo change 3.3e, whatever that was. . . . 292</p> <p>1994-05-10 ltdefs.dtx v1.0n</p> <p>General: (ASAJ) Added <code>\DeclareProtectedCommand</code>. . . 36</p>	<p>Added <code>\DeclareProtectedCommand</code> . . 44</p> <p>Removed braces around <code>\@ifundefined</code> argument. ASAJ. 41</p> <p><code>\makeatother</code>: Added <code>\makeatletter</code> and <code>\makeatother</code> ASAJ. 51</p> <p>1994-05-10 lterror.dtx v1.0n</p> <p><code>\@latexerr</code>: (ASAJ) Added extra blank lines to <code>\@latexerr</code>. . . . 65</p> <p>1994-05-10 ltmiscen.dtx v1.0j</p> <p><code>\@sverb</code>: Slight change in error message text. 348</p> <p>1994-05-11 ltboxes.dtx v1.0f</p> <p><code>\@begin@tempboxa</code>: Use new <code>\color@setgroup</code> concept. . . 381</p> <p><code>\@iiminipage</code>: Use new <code>\color@setgroup</code> concept. . . 388</p> <p><code>\@mpfootnotetext</code>: Use new <code>\color@setgroup</code> concept. . . 389</p> <p>Use new <code>\normalcolor</code> and <code>\@finalstrut</code>. 389</p> <p>General: Superfluous braces removed from several commands 380</p> <p><code>\color@setgroup</code>: macro added for colour support 382</p> <p><code>\endminipage</code>: Use new <code>\color@setgroup</code> concept. . . 388</p> <p>1994-05-11 ltclass.dtx v1.0c</p> <p><code>\endfilecontents</code>: Add checks for form feed and tab 571</p> <p>1994-05-11 ltdirchk.dtx v1.0e</p> <p>General: Add <code>\ProvidesFile</code> as used in fd files. 4</p> <p>1994-05-11 lterror.dtx v1.0o</p> <p><code>\@latexerr</code>: (ASAJ) Removed one of the extra blank lines to <code>\@latexerr</code>. 65</p> <p>1994-05-11 ltlogos.dtx v1.0o</p> <p><code>\LaTeX</code>: Use <code>\DeclareProtectedCommand</code>. ASAJ. 87</p> <p><code>\LaTeXe</code>: Use <code>\DeclareProtectedCommand</code>. ASAJ. 87</p> <p>1994-05-11 ltoutenc.dtx 1.5a</p> <p>General: Made T1 and OT1 generate packages rather than def files. Renamed the ‘package’ module to ‘teststy’. 105</p>
---	--

- 1994-05-11 ltoutenc.dtx v1.5a
 General: Reimplemented
`\DeclareTextCommand` using
`\@changed@cmd` and
`\DeclareProtectedCommand`. 106
 Renamed the commands again.
 Made the encoding part of the
 command syntax. Added the
`\DeclareTextCommand`
 interface. Used
`\DeclareProtectedCommand`. 102
`\DeclareTextAccent`:
 Reimplemented using
`\DeclareTextCommand`. 108
- 1994-05-11 ltspace.dtx v1.0o
`\,`: Use `\DeclareRobustCommand`.
 ASAJ. 85
`\hspace`: Use
`\DeclareRobustCommand`.
 ASAJ. 85
- 1994-05-12 ltboxes.dtx v1.0g
`\@finalstrut`: macro added 391
`\fbox`: New definition, merged
 with `\framebox` 384
`\framebox`: Merged `\fbox` and
`\framebox` 384
`\normalcolor`: macro added for
 colour support 382
- 1994-05-12 ltdefs.dtx v1.0p
 General: (ASAJ) Fixed a bug with
`\relax` which was using
`\@gobble` before defining it. .. 36
 Fixed a bug with `\relax` which
 was using `\@gobble` before
 defining it. 44
- 1994-05-12 ltssbas.dtx v2.1j
 General: New `baselinestretch`
 concept 153, 176
 Replaced hand-protected
 commands by
`\DeclareRobustCommand`
 defs 153, 176
`\f@linespread`: New macro 161, 184
`\fontencoding`: Use
`\DeclareRobustCommand`. 159, 182
`\fontfamily`: Use
`\DeclareRobustCommand`. 160, 183
`\fontseries`: Use
`\DeclareRobustCommand`. 160, 183
`\fontshape`: Use
`\DeclareRobustCommand`. 160, 183
`\fontsize`: Redefined to use
`\set@fontsize` 161, 184
`\linespread`: New macro .. 161, 184
- `\mathversion`: Use
`\DeclareRobustCommand`. 162, 185
- 1994-05-12 ltssdcl.dtx v2.1g
 General: Allow `\relax` as
 undefined command 226
 Allow `\relax`'ed cmds to be
 declared 226
- 1994-05-12 ltssini.dtx v2.1i
 General: Moved `\fontencoding` to
 fam.dtx 252
 Moved `\fontfamily` to fam.dtx 252
 Moved `\fontseries` to fam.dtx 252
 Moved `\fontshape` to fam.dtx 252
 Moved `\fontsize` to fam.dtx . 252
 Moved `\mathversion` to
 fam.dtx 252
 Moved `\selectfont` to
 tracefnt.dtx 252
- 1994-05-12 ltssstrc.dtx v2.3f
`\selectfont`: Use
`\DeclareRobustCommand` ... 203
- 1994-05-12 ltoutenc.dtx 1.5a
 General: Removed the
`\SaveAtCatcode` and
`\RestoreAtCatcode`
 commands. 118
 Rewrote for the new
 syntax. 118, 120
- 1994-05-12 ltoutput.dtx v1.0p
`\@writeseup`:
`\normalcoloradded` 508
 General: `\normalcoloradded` in
 various places (DPC). 481
- 1994-05-13 ltboxes.dtx v1.0h
`\@arrayparboxrestore`: New
 accent system, use `\let` not
`\def` 387
- 1994-05-13 ltcounts.dtx v1.0f
 General: Removed `\@Ialph` 149
 Removed `\@ialph` 149
- 1994-05-13 ltdefs.dtx v1.0q
 General: (ASAJ) Renamed
`\DeclareProtectedCommand` to
`\DeclareRobustCommand`.
 Removed
`\@if@short@command`. 36
 (ASAJ) Replaces `\space` by ‘ ’
 in `\csname`. 36
 Renamed
`\DeclareProtectedCommand` to
`\DeclareRobustCommand`.
 Removed `\@if@short@command`.
 Moved to after the definition of
`\@gobble`. 44

- 1994-05-13 ltdefs.dtx v1.0r
 General: (ASAJ) Added logging message to
 `\DeclareProtectedCommand`. . . 36
 Added logging message to
 `\DeclareProtectedCommand`. . . 44
- 1994-05-13 ltdefs.dtx v1.0s
 General: (ASAJ) Added
 `\@backslashchar`. 36
 (ASAJ) Coded `\@ifdefinable` more efficiently. 36
 Coded more efficiently, thanks to FMi. 41
- 1994-05-13 ltfiles.dtx LaTeX2e
 `\listfiles`: Stop `\listfiles` being run twice 100
- 1994-05-13 ltfiles.dtx v1.0g
 `\document`: Added execution of `\every@size` 91
- 1994-05-13 ltfinal.dtx v0.1h
 General: Added package `otlenc`, and defined `\@acci`, `\@accii` and `\@acciii`. 613
- 1994-05-13 ltfinal.dtx v1.0h
 General: Added output enc stuff . 625
- 1994-05-13 ltfloat.dtx v1.0g
 `\@footnotetext`: (DPC) Add new style colour support:
 `\normalcolor` 470
 (DPC) Use `\@finalstrut` 470
 `\@xfloat`: (DPC) Use
 `\normalcolor` 458
- 1994-05-13 ltfntcmd.dtx v3.3g
 General: Replaced `\@protecteddef` by `\DeclareRobustCommand` . 292
- 1994-05-13 ltfssbas.dtx v2.1k
 General: Remove File identification ‘typeout’ 153, 176
- 1994-05-13 ltfssbas.dtx v2.1l
 `\DeclareFontEncoding`: Init encoding change command 156, 179
 `\define@newfont`: Use `\@input@` for fd files 164, 187
- 1994-05-13 ltfssdcl.dtx v2.1h
 General: Removed file identification typeout 226
- 1994-05-13 ltfssini.dtx v2.1j
 General: Removed file identification typeout 252
- 1994-05-13 ltfssstrc.dtx v2.3g
 General: Removed typeouts as `\ProvidesPackage` writes to log. 199
- 1994-05-13 ltoutenc.dtx v1.5b
 General: Added `\{`, `\}` and `\$`. . . 102
 Renamed
 `\DeclareProtectedCommand` to `\DeclareRobustCommand`. . . . 102
 Replaces `\space` by ‘ ’ in
 `\csname`. 102
- 1994-05-13 ltpictur.dtx v0.1d
 General: Removed surplus braces from `\@if..` constructions . . 414
- 1994-05-13 lttab.dtx v1.0d
 `\@contfield`: Colour support . . 398
 `\@startfield`: Colour support . . 397
 `\@stopfield`: Colour support . . 398
 `\a`: moved to ltoutenc 396
- 1994-05-14 fontdef.dtx v2.1f
 General: Removed .def files. . . . 270
- 1994-05-14 ltfssbas.dtx v2.1m
 `\enc@update`: Macro added . . . 160, 183
- 1994-05-14 ltfssbas.dtx v2.1n
 General: Set defaults for all
 `\f@...` 161, 184
 `\DeclareErrorFont`: Don’t set
 `\f@encoding` 167, 190
 `\DeclareFontEncoding`: Log if encoding is redeclared . . 156, 179
 Only init enc change cmd when new encoding 156, 179
- 1994-05-14 ltfssini.dtx v2.1k
 General: Init error font just before checking for fontdef.cfg 266
 `\p@reset@font`: Remove surplus braces 265
- 1994-05-14 ltfssstrc.dtx v2.3h
 `\selectfont`: Added
 `\enc@update` 204
- 1994-05-14 ltoutenc.dtx 1.5d
 General: Moved the driver to the top. 105
- 1994-05-14 ltoutenc.dtx v1.5c
 General: Added the fontenc package 142
 Added the fontenc package. . . 102
 Fixed a bug which caused an infinite loop if `\f@encoding` was incorrectly set. . . . 102, 106
 Moved fontsmpl to its own dtx file. 102
- 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`. . 102, 106
 - Tidied up the documentation. . 102
- 1994-05-14 ltoutenc.dtx v1.5e
 - General: Replaced `\ENC@cmd` by `\ENC-cmd`. 102
- 1994-05-15 ltffsbas.dtx v2.1o
 - General: encoding cmds changed to enc-cmd 153, 176
- 1994-05-16 fontdef.dtx v2.1g
 - General: Removed `\DeclareFontEncoding` for ot1 and t1 and input .def files instead 270
- 1994-05-16 ltalloc.dtx v1.1a
 - General: (ASAJ) Split from ltinit.dtx. 54
- 1994-05-16 ltcntrl.dtx v1.0a
 - General: (ASAJ) Split from ltinit.dtx. 56
- 1994-05-16 ltdefns.dtx v1.1a
 - General: (ASAJ) Split from ltinit.dtx. 36
- 1994-05-16 lterror.dtx v1.1a
 - General: (ASAJ) Completely new error interface. 60
 - (ASAJ) Split from ltinit.dtx. . . 60
- 1994-05-16 ltfinal.dtx v1.0i
 - General: moved output enc stuff to lfonts 625
- 1994-05-16 ltffsbas.dtx v2.1p
 - `\fontsize`: Pass `\baselinestretch` not `\f@linespread` . . . 161, 184
 - `\linespread`: Remove surplus braces 161, 184
- 1994-05-16 ltffsini.dtx v2.1m
 - `\@acciii`: Define saved versions of accents 267
- 1994-05-16 ltlogos.dtx v1.1a
 - General: (ASAJ) Split from ltinit.dtx. 87
- 1994-05-16 ltmath.dtx v1.0k
 - `\ensuremath`: Use `\DeclareRobustCommand` and add extra braces in math mode 360
- 1994-05-16 ltoutenc.dtx 1.5h
 - General: `\pounds` was still using u rather than ui shape. 118
- 1994-05-16 ltoutenc.dtx v1.5f
 - General: enc files now have uc encoding name parts (FMi) . 102
- Revert code so that the encoding given is used in `\DeclareTextCommand` (FMi) 102
- 1994-05-16 ltoutenc.dtx v1.5g
 - General: Made fontenc.sty use the new mixed-case encoding files. 102
 - Removed the lowercasing of the filename. 142
- 1994-05-16 ltoutenc.dtx v1.5h
 - General: Added `\NG`, `\ng`, `\TH`, `\th`, `\DH`, `\dh`, `\DJ` and `\dj`. . . . 102
 - Added `\r` (ring accent) and `\k` (ogonek) accents. 102
 - Fixed a bug with `\pounds`. . . 102
 - Removed `\P` from the OT1 definitions file. 102
- 1994-05-16 ltoutenc.dtx v1.5i
 - General: Fixed a bug with `\d`. . . 102
- 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? 509
 - General: Changed setting of accents (FMi). 481
- 1994-05-16 ltpar.dtx v1.1a
 - General: (ASAJ) Split from ltinit.dtx. 69
- 1994-05-16 ltplain.dtx v1.0h
 - General: Comment out encoding specific commands 29
 - Remove `\@acci` and friends again 29
 - Remove unnecessary def for `\item` 28
 - `\loop`: Use Kabelschacht method 27
 - `\m@th`: Remove unnecessary space 28
- 1994-05-16 ltspc.dtx v1.1a
 - General: (ASAJ) Split from ltinit.dtx. 71
- 1994-05-17 ltclass.dtx v1.0e
 - `\@use@option`: Execute option after removing from list, not before 564
- 1994-05-17 ltdefns.dtx 1.1b
 - General: (ASAJ) Added the `\@protect@...` commands. . . 45
- 1994-05-17 ltdefns.dtx v1.1b
 - General: (ASAJ) Added definitions for protect. 36
 - (ASAJ) Removed warnings and logging to lterror.dtx. 36
 - Added the discussion of protected commands, defined

the values that <code>\protect</code> should have.	44	1994-05-19 ltdefs.dtx v1.1d	General: (RmS) Added definitions for <code>\@namedef</code> and <code>\@nameuse</code> again.	36
1994-05-17 ltdefs.dtx v1.1c		1994-05-19 ltfinal.dtx v0.1k	General: Removed <code>\makeat...</code> . . .	613
General: (ASAJ) Redid definitions for <code>protect</code>	36	1994-05-19 ltidxglo.dtx v1.1a	General: Initial version of <code>ltidxglo.dtx</code> , split from <code>ltidxbib.dtx</code>	472
1994-05-17 lterror.dtx v1.1b		1994-05-19 ltlength.dtx v1.1a	General: Extract file <code>ltlength</code> from <code>lctntlen</code>	152
General: (ASAJ) Moved error stuff from <code>ltdefs.dtx</code>	60	1994-05-19 ltpageno.dtx v1.1a	General: Extract file <code>ltpageno</code> from <code>lctntlen</code>	331
1994-05-17 ltfssini.dtx v2.1n		1994-05-19 ltplain.dtx v0.1k ltfinal	<code>\showoutput</code> : used <code>\maxdimen</code> not 99999	30
<code>\copyright</code> : Really add extra braces	265	<code>\showoverfull</code> : used <code>\One</code> not 1 . . .	30	
<code>\nfss@text</code> : Added braces to allow use in subscripts	265	1994-05-19 ltxref.dtx v1.1a	General: Extract file <code>ltxref</code> from <code>lctntlen</code>	332
1994-05-17 ltmath.dtx v1.0i		1994-05-20 ltdefs.dtx v1.1e	General: Changed command name from <code>\@checkcommand</code> to <code>\CheckCommand</code>	36
General: Replaced <code>\let</code> by <code>\gdef</code> , for indirect definition.	356	<code>\CheckCommand</code> : Changed name from <code>\@checkcommand</code> to <code>\CheckCommand</code>	43	
1994-05-17 ltoutenc.dtx v1.5j		1994-05-20 lterror.dtx v1.1c	General: (ASAJ) Added <code>\@latex@info@no@line</code>	60
General: Added braces to <code>\pounds</code> so it works as a subscript. . . .	102	(ASAJ) Added missing full stops.	60	
1994-05-18 ltdefs.dtx 1.1c		(ASAJ) Fixed a bug with <code>\@inmatherr</code>	60	
General: (ASAJ) Renamed the commands, and removed one which is no longer needed. . . .	45	1994-05-20 ltfinal.dtx v0.1l	General: Use new font warning commands	619
1994-05-18 ltdefs.dtx v1.1c		1994-05-20 ltfloat.dtx v1.0h	<code>\@endfloatbox</code> : Restore outer value of <code>@nobreak</code> switch. . . .	462
General: Redid the discussion and definitions, in line with the proposed new setting of <code>\protect</code> in the output routine.	44	<code>\outer@nobreak</code> : Macro added: default is to do nothing. . . .	462	
1994-05-18 ltfinal.dtx v0.1j		1994-05-20 ltfntcmd.dtx v3.3h	General: Use new error commands . . .	292
General: Corrected the lcode for d-bar.	613	1994-05-20 ltfssbas.dtx v2.1q	General: Use new error commands	153, 176
1994-05-18 ltlogos.dtx v1.1b		1994-05-20 ltfssstrc.dtx v2.3i	General: Use new error command names	199
General: (ASAJ) Added the \TeX logo.	87			
(ASAJ) Made the $\mathrm{L}^{\mathrm{T}}\mathrm{E}_{\mathrm{X}}2_{\epsilon}$ logo use the text font ‘2’ rather than the math font ‘2’.	87			
1994-05-18 ltoutenc.dtx v1.5k				
General: Made dotted-i produce ‘i’.	102			
Removed braces from <code>\pounds</code> and <code>\dollar</code>	102			
Replaced <code>\defaultencoding</code> with <code>\encodingdefault</code>	102			
1994-05-19 ltbibl.dtx v1.1a				
General: Initial version of <code>ltbibl.dtx</code> , split from <code>ltidxbib.dtx</code>	474			
1994-05-19 ltcounts.dtx v1.1a				
General: Extracted file from <code>lctntlen</code>	145			

1994-05-20 ltmiscen.dtx v1.0l	1994-05-22 lterror.dtx v1.2a
\@writefile: Added correct	General: (ASAJ) Made
setting of \protect. 340	\GenericError,
1994-05-20 ltmiscen.dtx v1.0m	\GenericWarning and
General: Use new warning	\GenericInfo robust. 60
commands 336	(ASAJ) Replaced
1994-05-20 ltoutput.dtx v1.0s	\@generic@message and
\@writsetup: Added setting of	\@generic@error by
\protect during \shipout. . 508	\GenericError,
General: Added setting of	\GenericWarning and
\protect during \shipout. . 481	\GenericInfo. 60
1994-05-20 ltpage.dtx v1.0d	(ASAJ) Replaced \ and tilde
\markright: Changed setting for	by \MessageBreak and \space. 60
\protect. 479	(ASAJ) Replaces \string by
1994-05-20 ltsect.dtx v1.0c	\protect in some messages. . 60
General: Correct setting of	1994-05-22 lterror.dtx v1.2d
\protect. 450	\GenericError: (DPC) Alternative
\addcontentsline: Correct setting	version added for old TeXs . . 60
of \protect. 450	(DPC) New version using long
1994-05-21 ltbibl.dtx v1.1b	command name. 60
General: Use new warning	1994-05-22 ltfloat.dtx v1.0i
commands 474	General: Use new warning
1994-05-21 lterror.dtx v1.1d	commands 453
General: (ASAJ) Made the error	1994-05-22 ltoutput.dtx v1.0t
commands robust. 60	General: Changed warnings and
1994-05-21 ltfiles.dtx v1.0h	infos to new commands. . . . 481
General: Use new error commands 88	1994-05-22 ltpictur.dtx v0.1e
1994-05-21 ltlists.dtx v1.0f	General: Use new warning cmds . 414
General: Use new error commands 364	1994-05-23 ltclass.dtx v1.0h
1994-05-21 ltmiscen.dtx v1.0n	\NeedsTeXFormat: Don't stop
General: Use new error commands 336	completely when format is
1994-05-21 ltsect.dtx v1.0d	wrong 566
General: Use new error commands 441	\usepackage: Remove argument if
1994-05-21 lttab.dtx v1.0f	possible 566
General: Use new error commands 392	1994-05-23 ltdirchk.dtx v1.0f
1994-05-21 ltxref.dtx v1.1b	General: Document \@TeXversion 1
General: Use new warning	1994-05-23 ltfstrc.dtx v2.3j
commands 332	General: Removed def of
\newlabel: Use new warning	\f@warn@break 216
commands 334	1994-05-23 ltoutput.dtx v1.0u
1994-05-22 ltclass.dtx v1.0f	\@activechar@info: Added
General: Use new warning and	\MessageBreak 507
error commands 553	\@writsetup: Changed resetting
1994-05-22 ltdefns.dtx v1.1f	of \protect after shipout to
General: Use new warning and	use \aftergroup 508
error cmds 36	General: Added \MessageBreak. . 481
1994-05-22 lterror.dtx v1.1e	Changed resetting of \protect
General: (ASAJ) Replaced bgroup	after shipout. 481
by begingroup in error	1994-05-24 lterror.dtx v1.2e
messages, to stop extra	\@latex@info@no@line: Macro
mathords creeping into math	added 63
mode. 60	1994-05-24 lterror.dtx v1.2f
	General: (DPC) wrap long lines . 60

1994-05-24 ltfnctcmd.dtx v3.3i		1994-05-31 ltfinal.dtx v1.0n	
General: Tidying and typos fixed	292	General: Renamed lthyphen.* to	
1994-05-24 ltmiscen.dtx v1.0q		lthyphen.*.	613
\@currentvline: Use \@empty as		1994-06-01 ltboxes.dtx v1.0i	
outer default	343	\@framebox: Macro added.	385
1994-05-25 ltdirchk.dtx v1.0g		\@ifframebox: New version, so	
\filename@parse: Mac parser had		\width is correct in	
" typo for :	12	\framebox	385
1994-05-25 ltfnctcmd.dtx v3.3j		\fbbox: New version, using	
General: Insertion of \aftergroups		\@framebox	384
to implement \nocorr moved		\framebox: New version, so \width	
to the end of the group	292	is correct in \framebox	384
\check@icr: Macros added	296	1994-06-01 ltlogos.dtx v1.1d	
\check@nocorr@: Insertion of		\LaTeX: Add \m@th to force math	
\aftergroups moved and		size calculations	87
defaults set up for efficiency .	296	1994-06-01 ltoutput.dtx v1.0w	
\DeclareTextFontCommand:		General: Tidied up typesetting. .	481
\expandafter inserted	294	1994-06-08 ltfinal.dtx v1.0m	
Insertion of \aftergroups		General: Add patch file system . .	625
moved	294	1994-06-09 ltfinal.dtx v1.0n	
1994-05-25 ltoutput.dtx v1.0v		General: For T _E X ₂ , do not set	
General: Extra documentation. . .	481	codes for higher half of	
1994-05-25 ltsect.dtx v1.0e		character table.	617, 623
\@dottedtocline: Put braces		1994-06-09 ltfnctcmd.dtx v3.3k	
around argument 4 (the actual		General: Tidying and typos fixed	
toc entry) to avoid font (and		in documentation	292
possibly other) changes leaking		1994-06-18 ltfnctcmd.dtx v3.3l	
out to the leaders.	451	General: Added check for empty	
1994-05-25 ltthm.dtx v1.0c		text	292
General: Modify documentation . .	437	\check@nocorr@: Added check for	
1994-05-25 ltvers.dtx v1.0d		empty text	296
General: Remove PRELIMINARY		1994-06-22 ltfnctcmd.dtx v3.3m	
TEST RELEASE from startup		General: Removed space from	
banner (spring is here)	33	\nfss@text	292
1994-05-25 ltxref.dtx v1.1c		Renamed \check@nocorr	292
General: Modify documentation . .	332	\check@nocorr@: Renamed	
1994-05-26 ltfiles.dtx LaTeX2e		\check@nocorr to	
\@missingfileerror: Modify		\text@command to improve	
message format	99	\long error message	296
1994-05-26 ltlogos.dtx v1.1c		\DeclareTextFontCommand:	
General: Remove \SLiTeX logo . .	87	Removed space from	
1994-05-26 ltmiscen.dtx v1.0r		\nfss@text	294
General: \literal removed	350	1994-06-22 ltmath.dtx v1.2t classes	
1994-05-26 ltplain.dtx v1.1m		\mathindent: Set \mathindent at	
\iterate: (CAR) added \long . .	27	the end of the class instead of	
\underbar: (CAR/FMi) changed		at begin document	361
to use box \tw@	28	1994-07-20 ltlogos.dtx v1.1e	
1994-05-26 ltplain.dtx v1.1p		\LaTeX: Save a few tokens	87
\underbar: (DPC) changed to use		\LaTeXe: Save a few tokens	87
\abox	28	1994-07-20 ltpage.dtx v1.0h	
1994-05-29 ltffsdel.dtx v2.1j		\sloppy: Save a few tokens	480
General: Use new error commands	226	1994-09-16 ltffsbas.dtx v2.1s	
		\nfss@catcodes: Reset [and] as	
		well, just in case	165, 188

- 1994-10-07 ltoutenc.dtx v1.5l
General: Moved the ogonek accent. 102
- 1994-10-11 ltdirchk.dtx v1.0h
`\TeXversion`: Check for TeX3.14 13
General: Modify all of ltxcheck again 14
- 1994-10-12 ltsect.dtx v1.0f
General: Doc. typos 441
- 1994-10-14 fontdef.dtx v2.2a
General: New coding 268
- 1994-10-14 ltfsini.dtx v2.2a
General: New coding for cfg files . 252
- 1994-10-14 ltmiscen.dtx v1.0s
General: Move math to other file 336
- 1994-10-14 ltplain.dtx v1.1a
General: Moved code to other files. 15
- 1994-10-15 ltfsbas.dtx v2.1t
`\extract@alph@from@version`: Warn if math alpha is used outside math 172, 195
- 1994-10-18 ltboxes.dtx v1.0j
`\@frameb@x`: `\leavevmode` added 385
`\@ifframebox`: `\leavevmode` moved to `\@frameb@x` 385
`\@parboxto`: Macro added to remove misuse of `\@empty` . . 386
General: stuff from ltpatch done . 380
`\fbox`: `\long` added 384
`\mbox`: `\long` added 381
`\sbox`: `\long` added 383
- 1994-10-18 ltclass.dtx v1.0j
General: Move `\listfiles` to ltfiles.dtx 553
- 1994-10-18 ltdefns.dtx v1.2a
`\@star@or@long`: macro added . . 38
General: Add extra test for `\endgraf` 36
Add star-forms for all commands `\renew@environment`: reset end command 42
- 1994-10-18 ltfiles.dtx v1.0i
`\listfiles`: code moved here from ltclass 100
- 1994-10-18 ltoutenc.dtx v1.5l
General: Added new definitions of `\patterns` and `\hyphenation`. 113
- 1994-10-18 ltoutenc.dtx v1.5m
General: Added new definitions of `\patterns` and `\hyphenation`. 102
- 1994-10-18 ltsect.dtx v1.0g
`\@dottedtocline`: Added `\normalcolor` for page number 451
- General: Added `\normalcolor` . . 441
- 1994-10-19 ltfsbas.dtx v2.1t
`\DeclareFontEncoding`: Add missing `\relax`. 155, 178
- 1994-10-23 ltfsstrc.dtx v23.k
`\every@math@size`: Renamed to `\every@math@size` 206
- 1994-10-23 ltmath.dtx v1.0l
`\@eqnnum`: Added `\normalcolor` since `\eqno` introduces a subgroup of the displayed math group 358
- `\ensuremath`: Remove extra braces: but see p 168 of Leslie's book 360
- 1994-10-24 ltboxes.dtx v1.0k
`\fbox`: Inner braces added (to fix latex/1061) 384
- 1994-10-25 fontdef.dtx v2.2c
General: Added OMSenc.def . . . 270
- 1994-10-25 ltboxes.dtx v1.0l
`\@isavepicbox`: missing percent (moved from ltpatch) 383
- 1994-10-25 ltdefns.dtx v1.2b
General: Documentation improvements 36
- 1994-10-25 ltoutenc.dtx 1.6a
General: Added `\textdollar`, `\textlbrace`, `\textrbrace`, `\textsterling`, `\textunderline`. 120
Removed `\textlbrace`, `\textrbrace`, `\textunderline` to give them their proper names. 120
- 1994-10-25 ltoutenc.dtx v1.6a
General: Added `\ProvideTextCommand`, `\UseTextSymbol`, `\UseTextAccent`, `\DeclareTextSymbolDefault`, `\DeclareTextAccentDefault`, `\DeclareTextCommandDefault`, and `\ProvideTextCommandDefault`. 102
Added the `\Provide` commands, and the default definitions. . 106
Added the defaults. 114
Added the files OT1enc.def, T1enc.def and OMSenc.def. . 114
Added the OMS encoding. . . . 125
- 1994-10-27 ltoutenc.dtx 1.6b
General: Added `\textasciicircum` `\textasciitilde`

<code>\textbackslash \textbar</code>	1994-10-29 ltoutenc.dtx v1.6c
<code>\textbraceleft</code>	General: Added commands like
<code>\textbraceright</code>	<code>\dots</code> for use in text and
<code>\textcompwordmark</code>	math. 114
<code>\textemdash \textendash</code>	Renamed <code>\P</code> , <code>\S</code> , <code>\dag</code> and
<code>\textexclamdown</code>	<code>\ddag</code> to <code>\textparagraph</code> ,
<code>\textgreater</code>	<code>\textsection</code> , <code>\textdagger</code>
<code>\texthyphenchar \texthyphen</code>	and <code>\textdaggerdbl</code> 102
<code>\textless \textquestiondown</code>	1994-10-30 ltdefns.dtx v1.2c
<code>\textquotedblleft</code>	<code>\@onelevel@sanitize</code> : Macro
<code>\textquotedblright</code>	added 51
<code>\textquotedbl</code>	General:
<code>\textquoteleft</code>	<code>(CAR)\@onelevel@sanitize</code>
<code>\textquoteright</code>	added 36
<code>\textunderscore</code>	1994-10-30 ltdefns.dtx v1.2f
<code>\textvisiblespace</code> 120	General: <code>(DPC)\newwrite</code> 's moved
Added: <code>\textemdash</code>	to ltfiles 36
<code>\textendash \textexclamdown</code>	1994-10-30 ltmath.dtx v1.0n
<code>\texthyphenchar \texthyphen</code>	General: ASAJ: Moved the new
<code>\textquestiondown</code>	commands to ltoutenc. 356
<code>\textquotedblleft</code>	1994-10-30 ltoutenc.dtx v1.6d
<code>\textquotedblright</code>	General: Added <code>\DeclareTextCom-</code>
<code>\textquoteleft</code>	<code>positeCommand</code> 102
<code>\textquoteright</code> 118	Added <code>\textcircled</code> . 102, 115, 125
1994-10-27 ltoutenc.dtx v1.5d	Added <code>\t</code> 115
General: Rewrote	Added math commands. 102
<code>\DeclareTextSymbol</code> to define	Added OML encoding. . . 102, 115
its argument to use the current	Added the OML encoding. . . 126
encoding by default, to fit with	Made <code>\textless</code> and
<code>\DeclareTextCommand</code> 106	<code>\textgreater</code> come from
1994-10-27 ltoutenc.dtx v1.6b	OML. 115
General: Added <code>\textbackslash</code> . 125	Moved math commands here
Added more defaults for OT1. 114	from ltmath. 118
Removed the enc.def files 102	Removed <code>\textregistered</code> . . . 115
Removed the files OT1enc.def,	Rewrote <code>\copyright</code> to use
OT1enc.def and OMSenc.def. . . 114	<code>\textcircled</code> 115
Renamed <code>\textlbrace</code> to	1994-10-31 fontdef.dtx v2.2d
<code>\textbraceleft</code> and	General: Added OMLenc.def 270
<code>\texttrbrace</code> to	1994-10-31 fontdef.dtx v2.2e
<code>\textbraceright</code> 125	General: ... and moved further
1994-10-29 ltmath.dtx 1.0m	down 270
General: ASAJ: Added	1994-10-31 ltfloating.dtx v1.1a
<code>\DeclareMathOperator</code> 351	<code>\@dblfloat</code> : Major changes since
ASAJ: Tidied up	two-column and one-column
documentation. 357	cases merged 457
1994-10-29 ltmath.dtx v1.0m	<code>\@dblflset</code> : Macro added 456
General: ASAJ: Added	Major changes to parameter
<code>\mathellipsis</code> , <code>\mathdollar</code>	parsing, setting of local
and <code>\mathsterling</code> 356	variables, etc; two-column and
ASAJ: Removed <code>\dag</code> , <code>\ddag</code> . . 356	one-column cases merged;
ASAJ: Renamed <code>\S</code> and <code>\P</code> to	space hacks moved 456
<code>\mathsection</code> and	<code>\@endfloatbox</code> : (DPC/CAR)
<code>\mathparagraph</code> and made	Extra box added to remove
them <code>\mathchardefs</code> 356	colour resetting from vmode . 462

<code>\@floatboxreset</code> : Macro added . 460	Rewrote protected short commands using <code>\x@protect</code> . ASAJ. 45
<code>\@footnotetext</code> : (DPC/CAR) Move colour setting to output routine 470	1994-11-04 <code>lterror.dtx</code> v1.2g General: Added
<code>\@savemarbox</code> : (DPC/CAR) Extra box added for colour 465	<code>\set@display@protect</code> to <code>\Generic*</code> commands. ASAJ. 60
<code>\@setfps</code> : Macro added 457	1994-11-04 <code>ltfiles.dtx</code> v1.0k
<code>\@xdblfloat</code> : Macros removed: <code>\@dbflt</code> , <code>\@xdblfloat</code> 462	<code>\nofiles</code> : Added setting of <code>\protected@write</code> , <code>\makeindex</code> and <code>\makeglossary</code> to <code>\nofiles</code> . ASAJ. 93
<code>\@xfloat</code> : (DPC/CAR) Extra box added to remove colour resetting from <code>vmode</code> 458	<code>\protected@write</code> : Macro added ASAJ. 94
Major changes, removing setting of local variables, space hacks etc; two-column and one-column cases merged . . . 457	1994-11-04 <code>ltfloat.dtx</code> v1.1b
Reset hook added 458	<code>\@footnotetext</code> : (ASAJ) Added <code>\protected@edef</code> 470
<code>\@xympar</code> : (DPC/CAR) Extra box added since needed for floats 466	<code>\footnotemark</code> : Added <code>\protected@xdef</code> to <code>\footnotemark</code> 471
<code>\fps@dbl</code> : Macro added 457	1994-11-04 <code>ltidxglo.dtx</code> v1.1b
1994-10-31 <code>ltoutput.dtx</code> v1.1a	<code>\wrglossary</code> : Added <code>\protected@write</code> to <code>\wrglossary</code> 473
<code>\@makecol</code> : (DPC/CAR) Colour resetting moved to here 504	<code>\wrgindex</code> : Added <code>\protected@write</code> to <code>\wrgindex</code> 473
<code>\@topnewpage</code> : (DPC/CAR) Extra box added to remove colour resetting from <code>vmode</code> 496	General: Removed <code>\if@files</code> from <code>\makeindex</code> 472
(DPC/CAR) Use <code>\color@begingroup</code> for colour 496	<code>\makeglossary</code> : Removed <code>\if@files</code> from <code>\makeglossary</code> 473
(DPC/CAR) Use <code>\normalcolor</code> 496	1994-11-04 <code>ltmiscen.dtx</code> v1.0t
1994-11-02 <code>ltoutenc.dtx</code> v1.6d	<code>\@writefile</code> : Removed setting of <code>\protect</code> . ASAJ. 340
General: Wrapped lines longer than 70 characters. 102	1994-11-04 <code>ltoutenc.dtx</code> v1.6f
1994-11-03 <code>ltclass.dtx</code> v1.0k	General: Added <code>_</code> 116
General: Move <code>\@missingfileerror</code> to <code>ltfiles</code> 557	Added <code>\mathunderscore</code> 118
1994-11-03 <code>ltdirchk.dtx</code> v1.0i	1994-11-04 <code>ltpage.dtx</code> v1.0e
General: Generate an error if <code>latex.ltx</code> not used with <code>clean initex</code> 1	<code>\markright</code> : Added <code>\@unexpandable@protect</code> . ASAJ. 479
1994-11-03 <code>ltfiles.dtx</code> v1.0j	1994-11-04 <code>ltsect.dtx</code> 1.0h
<code>\@missingfileerror</code> : Move here from <code>ltclass</code> 99	<code>\@sect</code> : (ASAJ) Added <code>\protected@edef</code> 445
1994-11-04 <code>ltboxes.dtx</code> v1.0m	General: (ASAJ) Added <code>\protected@xdef</code> to <code>\thanks</code> . 441
<code>\@mpfootnotetext</code> : Added <code>\protected@edef</code> . ASAJ. . . 389	1994-11-04 <code>ltsect.dtx</code> v1.0h
1994-11-04 <code>ltdefns.dtx</code> v1.2e	General: Added <code>\protected@write</code> to <code>\addtocontents</code> . ASAJ. . 450
General: Added <code>\set@display@protect</code> to <code>\typeout</code> . ASAJ. 36	<code>\addcontentsline</code> : Added <code>\protected@write</code> to <code>\addcontentsline</code> . ASAJ. . 450
Added commands for setting and restoring <code>\protect</code> . ASAJ. 46	

1994-11-04 lttab.dtx v1.0h		\@writesetup: Change protect settings for new-style, protect-free aux-files.	508
\@mkpream: (ASAJ) Added		Use new \color@hbox concept.	508
\@unexpandable@protect to		1994-11-05 ltoutput.dtx v1.1c	
\@mkpream.	410	\@beginndvi: Added macro	512
\multicolumn: (ASAJ) added		\@beginndvibox: Added macro	493
\set@typeset@protect.	406	\@writesetup: Add new	
1994-11-04 ltxref.dtx v1.1d		\AtBeginDvi concept	508
\label: (ASAJ)Added		\AtBeginDvi: Added macro	493
\protected@edef	334	1994-11-06 ltssbas.dtx v2.1u	
(ASAJ)Added		\cf@encoding: New macro	161, 184
\protected@write	334	\DeclareFixedFont: Renamed	
1994-11-05 ltboxes.dtx v1.0n		\every@size to	
\@mpfootnotetext: Colour		\every@math@size.	154, 177
resetting for footnotes moved		1994-11-06 ltssini.dtx v2.2b	
to endminipage: as for main		\@setsize: Use	
page.	389	\@typeset@protect	265
\color@endbox: macro added for		1994-11-06 ltssstrc.dtx v2.3k	
colour support	382	\glb@currsiz: New	
\color@hbox: macro added for		implementation	205
colour support	382	\try@simples: New	
\endminipage: Colour resetting for		implementation	216
footnotes moved to here: as for		\try@size@substitution: New	
main page.	388	implementation	216
1994-11-05 ltboxes.dtx v1.0o		\tryis@simple: New	
\@mpfootnotetext: Colour groups		implementation	216
restored here.	389	1994-11-07 fontdef.dtx v2.2f	
1994-11-05 ltfloat.dtx v1.1c		General: (DPC) Add	
\@dblflset: Add compatibility		\DeclareMathSizes	
with old version of \@xfloat.	456	declarations	275
\@endfloatbox: Use new		(DPC) Updated to use	
\color@hbox concept.	462	\ProvidesFile	270
\@footnotetext: Removed		1994-11-07 ltfiles.dtx v1.0l	
\normalcolor (again)	470	\@unused: move here from ltdefs,	
\@savemarbox: Use new		remove duplicate \@mainaux	90
\color@hbox concept.	465	1994-11-07 ltfiles.dtx v1.0m	
\@setfps: Add compatibility with		\document: Renamed \every@size	
old version of \@xfloat.	457	to \every@math@size.	91
\@xfloat: Add compatibility with		1994-11-07 preload.dtx v2.1e	
old version of \@xfloat: but		General: (DPC) Updated to use	
the arguments, provided at		\ProvidesFile	288
exorbitant cost, are now		1994-11-09 ltboxes.dtx v1.0p	
completely ignored	457	\@finalstrut: Revert	
Use new \color@hbox concept.	458	\finalstrut to 2.09 equivalent	
\@xympar: Use new \color@hbox		(from ltpatch)	391
concept.	466	General: more colour changes.	380
1994-11-05 ltoutenc.dtx v1.6g		1994-11-09 ltssbas.dtx v2.1v	
General: Added setting of		\@vpt: (DPC) macros added, from	
\@typeset@protect to		setsize.dtx	175, 198
\patterns and \hyphenation.	113	(DPC) reduce save stack usage	
1994-11-05 ltoutput.dtx v1.1b		latex/1742	175, 198
\@topnewpage: Use new		1994-11-10 ltbibl.dtx v1.1c	
\color@hbox concept.	496	General: Fix \nocite{*}	474

<code>\nocite</code> : Fix <code>\nocite{*}</code>	476	1994-11-14 <code>ltspace.dtx</code> v1.2d	
1994-11-10 <code>ltmath.dtx</code> v1.2v classes		<code>\:</code> (DPC) Macro modified	75
<code>eqnarray</code> : Added value of		1994-11-14 <code>lftab.dtx</code> v1.0i	
<code>\parskip</code> to		<code>\tabularnewline</code> : (DPC) Macro	
<code>\abovedisplayskip</code> to		added	405
compensate for negative		1994-11-16 <code>fontdef.dtx</code> v2.2h	
<code>\topsep</code>	362	General: (DPC) Removed <code>\{</code> and	
1994-11-10 <code>ltoutput.dtx</code> v1.1e		<code>\}</code>	270
<code>\@writesetup</code> : Modify <code>\protect</code>		1994-11-17 <code>ltboxes.dtx</code> v1.0q	
setting	508	General: <code>\@tempa</code> to <code>\reserved@a</code>	380
1994-11-10 <code>ltplain.dtx</code> v1.1b		1994-11-17 <code>ltclass.dtx</code> v1.0l	
General: (CAR) added patch to		General: <code>\@tempa</code> to <code>\reserved@a</code>	553
<code>\loop</code>	15	1994-11-17 <code>ltntrl.dtx</code> v1.0b	
<code>\iterate</code> : (CAR) added extra		General: <code>\@tempa</code> to <code>\reserved@a</code>	56
<code>\relax</code>	27	1994-11-17 <code>ltdefns.dtx</code> v1.0g	
1994-11-11 <code>ltspace.dtx</code> v1.2a		General: <code>\@tempa</code> to <code>\reserved@a</code>	36
<code>\:</code> (DPC) Make robust	75	1994-11-17 <code>ltdirchk.dtx</code> v1.0j	
1994-11-12 <code>lftncmd.dtx</code> v3.3o		General: <code>\@tempa</code> to <code>\reserved@a</code>	1
<code>\normalsize</code> : Added		1994-11-17 <code>lterror.dtx</code> v1.2h	
<code>\MessageBreak</code>	300	General: <code>\@tempa</code> to <code>\reserved@a</code>	60
1994-11-12 <code>ltlists.dtx</code> v1.2b <code>ltspace</code>		1994-11-17 <code>ltfiles.dtx</code> v1.0n	
<code>\endtrivlist</code> : Changed order of		General: <code>\@tempa</code> to <code>\reserved@a</code>	88
tests to make <code>\@noitemerror</code>		1994-11-17 <code>ltfinal.dtx</code> v1.0o	
correct: end of an era.	373	General: <code>\@tempa</code> to <code>\reserved@a</code>	613
1994-11-12 <code>ltmiscen.dtx</code> v1.0u		1994-11-17 <code>ltfloat.dtx</code> v1.1e	
<code>center</code> : Changed end macro to		General: <code>\@tempa</code> to <code>\reserved@a</code>	453
<code>\def</code> : safer and consistent . .	345	1994-11-17 <code>lftncmd.dtx</code> v3.3p	
<code>flushleft</code> : Changed end macro to		General: <code>\@tempa</code> to <code>\reserved@a</code>	292
<code>\def</code> : safer and consistent . .	345	1994-11-17 <code>ltfssbas.dtx</code> v2.1w	
<code>flushright</code> : Changed end macro		General: <code>\@tempa</code> to	
to <code>\def</code> : safer and consistent	345	<code>\reserved@a</code>	153, 176
1994-11-12 <code>ltplain.dtx</code> v1.1c		1994-11-17 <code>ltfssdcl.dtx</code> v2.1m	
General: Comment out more		General: <code>\@tempa</code> to <code>\reserved@a</code>	226
encoding specific commands . .	29	1994-11-17 <code>ltfssstrc.dtx</code> v2.3l	
1994-11-12 <code>ltspace.dtx</code> v1.2b		General: <code>\@tempa</code> to <code>\reserved@a</code>	199
<code>\addpenalty</code> : Corrected error		1994-11-17 <code>ltmath.dtx</code> v1.0o	
message	81	General: <code>\@tempa</code> to <code>\reserved@a</code>	351
<code>\addvspace</code> : Corrected error		1994-11-17 <code>ltmiscen.dtx</code> v1.0v	
message	81	General: <code>\@tempa</code> to <code>\reserved@a</code>	336
1994-11-13 <code>ltspace.dtx</code> v1.2c		1994-11-17 <code>ltoutenc.dtx</code> v1.6h	
<code>\addpenalty</code> : Recorrected error		General: (DPC) <code>\@tempa</code> to	
message	81	<code>\reserved@a</code>	102
<code>\addvspace</code> : Recorrected error		1994-11-17 <code>ltoutput.dtx</code> v1.1h	
message	81	General: <code>\@tempa</code> to <code>\reserved@a</code>	481
1994-11-14 <code>ltoutput.dtx</code> v1.1f		1994-11-17 <code>ltpictur.dtx</code> v1.0f	
<code>\@begindvi</code> : Use normal box		General: <code>\@tempa</code> to <code>\reserved@a</code>	414
register: why a box?	512	1994-11-17 <code>ltsect.dtx</code> v1.0i	
<code>\@begindvibox</code> : Use normal box		General: <code>\@tempa</code> to <code>\reserved@a</code>	441
register: why a box?	493	1994-11-17 <code>lftab.dtx</code> v1.0j	
<code>\@writesetup</code> : Modify new		General: <code>\@tempa</code> to <code>\reserved@a</code>	392
<code>\AtBeginDvi</code> concept	508	1994-11-18 <code>ltboxes.dtx</code> v1.0r	
General: Removed old definition of		<code>\color@vbox</code> : macro added for	
<code>\@testfp</code>	481	colour support	382

- 1994-11-18 ltfinal.dtx v1.0n
General: re-allow slots 127–255 . . . 623
- 1994-11-18 ltffsbas.dtx v2.1x
General: (DPC) use `\reserved@f`
not `\next` 153, 176
- 1994-11-18 ltffsdel.dtx v2.1m
`\DeclareMathDelimiter`: (DPC)
`\expandafter` instead of
`\next` 243
- 1994-11-18 ltffsstrc.dtx v2.3m
General: `\next` to `\reserved@f` . . . 199
- 1994-11-18 ltmath.dtx v1.0p
`\phantom`: (DPC) colour support . . . 353
(DPC) use `\expandafter`
instead of `\next` 353
`\prime@s`: (DPC) use `\@let@token`
instead of `\next` and
`\expandafter` instead of `\nxt` . . . 356
`\smash`: (DPC) colour support . . . 354
(DPC) use `\expandafter`
instead of `\next` 354
- 1994-11-21 ltfloat.dtx v1.1f
`\@endfloatbox`: Added reset of
minipage flag 462
Corrected position of
`\outer@nobreak` 462
`\@marginparreset`: Macro added . . . 465
`\@savemarbox`: Added
`\@setminipage` etc 465
Added resetting of size and font . . . 465
Changed to `\color@vbox` 465
Use `\@setnobreak` etc 465
`\@setminipage`: Macro added 460
`\@setnobreak`: Macro added 460
`\@xfloat`: Added `\@setminipage` . . . 458
Added resetting of size and font . . . 458
Changed to `\color@vbox` so
that large floats overflow at the
bottom 458
Missing percents reinserted after
4, 8: these are not numbers. 457
Use `\@setnobreak` 458
`\@xympar`: Changed to
`\color@vbox` 466
- 1994-11-21 ltoutput.dtx v1.1i
`\@addtocurcol`: Added
`\if@nobreak` test before float
box 521, 525
`\@specialoutput`: Added
`\if@nobreak` test 501
`\@topnewpage`: Changed to
`\color@vbox` 496
- 1994-11-22 ltffsdel.dtx v2.1o
General: wrap long lines 226
- 1994-11-22 ltoutenc.dtx v1.6i
General: Corrected `\dots` so that
there's no kerning in
monowidth fonts. 102
Corrected typo with
`\mathunderscore`. 102
Fixed empty accents. Again. 102
- 1994-11-24 ltdefs.dtx v1.2h
`\@newenv`: Added test for `\endgraf` . . 42
- 1994-11-25 ltplain.dtx v1.1f
General: (DPC) Comment out lots
of obsolete code 15
- 1994-11-26 ltfloat.dtx v1.1b
`\footnote`: (ASAJ) Added
`\protected@xdef`. 470
- 1994-11-28 ltcntrl.dtx v1.0c
General: Documentation
improvements 56
- 1994-11-30 ltfiles.dtx v1.0o
`\@dofilelist`: Macro added 101
`\listfiles`: Use `\@dofilelist` . . . 100
`\nofiles`: There is no
`\@gobblethree`... 93
- 1994-11-30 ltffsbas.dtx v2.1y
`\fontshape`: Use `\@current@cmd` in
`\@@enc@update`. ASAJ. 160, 183
- 1994-11-30 ltmath.dtx 1.0q
General: ASAJ:
`\DeclareMathOperator` moved
to AMSI^ATeX. 351
- 1994-11-30 ltmiscen.dtx v1.0w
`\enddocument`: (DPC) Do
warnings even for `\nofiles` . . . 338
(DPC) Use `\@dofilelist` 338
- 1994-11-30 ltoutenc.dtx 1.7a
General: Redefined `\a` for the new
scheme. 114
- 1994-11-30 ltoutenc.dtx v1.6g
General: Removed new definitions
of `\patterns` and
`\hyphenation`, since
encoding-specific commands
now expand in the mouth. 113
- 1994-11-30 ltoutenc.dtx v1.7a
General: Added new code for
encoding-specific commands.
These now expand in the
mouth, which means that
ligaturing and kerning can
happen. 102
Always load the enc.def file, so
that the default encoding for
the commands will change. 142

Redefined <code>\@changed@cmd</code> to expand in the mouth.	106	<code>\@dblflset</code> : Macro removed temporarily	456
Removed <code>\@changed@x@mouth</code> since <code>\@changed@x</code> now expands in the mouth.	106	Old version reinstated temporarily	456
Rewrote <code>\@text@composite</code> so it allows an empty argument, or an argument containing lots of commands.	108	<code>\@setfps</code> : Macro removed temporarily	457
1994-12-01 <code>ltfinal.dtx</code> v1.0p		<code>\@xdblfloat</code> : Macros reinserted temporarily	462
General: Renamed <code>lthyphen.*</code> to <code>hyphen.*</code>	613	<code>\@xfloat</code> : Old version reinstated temporarily	457
1994-12-01 <code>lthyphen.dtx</code> v1.0g		Sanitisation added temporarily	457
General: Rename <code>lthyphen.ltx/cfg</code> to <code>hyphen.ltx/cfg</code>	586	General: Some temps reinserted temporarily	453
1994-12-01 <code>ltplain.dtx</code> v1.1g		<code>\fps@dbl</code> : Macro removed temporarily	457
General: (DPC) More doc changes	15	1994-12-10 <code>lftncmd.dtx</code> v3.3q	
1994-12-02 <code>fontdef.dtx</code> v2.2i		<code>\@@math@egroup</code> : Don't read arguments	299
General: Commented out <code>\ldots</code> . ASAJ.	268	<code>\check@nocorr@</code> : Use <code>\space</code> command for comparison . . .	296
1994-12-02 <code>ltfssini.dtx</code> v2.2c		1994-12-10 <code>ltfssdcl.dtx</code> v2.1p	
<code>\copyright</code> : <code>\copyright</code> is now in <code>ltoutenc</code> . ASAJ	265	<code>\document@select@group</code> : Surround with braces (add fourth arg)	230
1994-12-02 <code>ltlists.dtx</code> v1.0e		<code>\select@group</code> : Surround with braces (add fourth arg)	228
<code>\@trivlist</code> : RmS: Added check for looping	372	1994-12-10 <code>ltoutenc.dtx</code> v1.7e	
1994-12-02 <code>ltoutenc.dtx</code> 1.7b		General: Added documentation for the OML encoding.	102
General: Redefined <code>\a</code> properly. .	114	Replaced width with <code>\@width</code> and ditto height in <code>vrules</code> . . .	102
1994-12-02 <code>ltoutenc.dtx</code> v1.7b		1994-12-14 <code>ltoutenc.dtx</code> v1.7f	
General: Fixed a bug with <code>\a</code> . . .	102	General: Added braces to <code>\copyright</code> so it works unbraced in subscripts. . . .	102
1994-12-04 <code>lthyphen.dtx</code> v1.0h		Added check for math mode in <code>\@changed@cmd</code>	102
General: Documentation edits for /1989	586	Commented out <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textbackslash</code> , <code>\textbar</code> , <code>\textgreater</code> , <code>\texthyphenchar</code> , <code>\texthyphen</code> and <code>\textless</code> to save memory.	102
1994-12-05 <code>ltoutenc.dtx</code> v1.7c		1995-01-12 <code>ltmath.dtx</code> v1.2y classes	
General: Added braces to <code>\textcircled</code>	102	<code>\eqnnum</code> : Added <code>\normalcolor</code> .	361
1994-12-06 <code>ltfssbas.dtx</code> v2.1z		1995-03-03 <code>ltoutenc.dtx</code> 1.7g	
<code>\DeclareFontEncoding</code> : use <code>\nfss@catcodes</code>	155, 178	General: Corrected an error in documentation referring to the tabular rather than the tabbing environment.	114
<code>\nfss@catcodes</code> : Added tab char as well	165, 188		
1994-12-08 <code>ltoutenc.dtx</code> v1.7d			
General: Added <code>\null</code> and <code>\sh@ft</code> to <code>\b</code> and <code>\d</code>	102		
1994-12-08 <code>lttab.dtx</code> v1.0k			
<code>\@array</code> : Add <code>\tabularnewline</code> .	404		
<code>\tabularnewline</code> : (DPC) Made it <code>\relax</code>	405		
1994-12-09 <code>ltbibl.dtx</code> v1.1d			
<code>\bibliographystyle</code> : (DPC) Allow use in preamble. . . .	476		
1994-12-10 <code>ltfloat.dtx</code> v1.1g			
<code>\@dblfloat</code> : Old version reinstated temporarily	457		

1995-04-02 ltfnctcmd.dtx v3.3r	1995-04-24 ltplain.dtx v1.1h
\backslash @math@egroup: Read them again	\backslash newlanguage: Remove remaining
to be able to add \backslash relax.	\backslash outer declarations. 17
1995-04-02 ltfssdcl.dtx v2.1q	1995-04-24 ltxref.dtx v1.1e
\backslash document@select@group: fix	\backslash newlabel: Make \backslash @onlypreamble
problem for pr/1275	for /1388. 334
\backslash select@group: fix problem for	1995-04-25 ltdefns.dtx v1.2i
pr/1275	\backslash @check@c: Make \backslash long for
\backslash set@mathdelimater: fix pr/1329	latex/1346 43
1995-04-02 ltfssini.dtx v2.2d	\backslash new@environment: Parse
\backslash not@math@alphabet: add	arguments slowly but safely
\backslash noexpand to second part of	/1507 41
message	1995-04-25 ltfiles.dtx v1.0q
1995-04-21 ltclass.dtx v1.0m	\backslash document: Removed execution of
\backslash DeclareOption*: Made long	\backslash every@size latex/1407 91
/1498	1995-04-25 ltsect.dtx v1.0k
\backslash endfilecontents: Close input	\backslash @dottedtocline: Added \backslash hbox
check stream: latex/1487 . . .	around dots. 451
1995-04-21 ltfinal.dtx v1.0q	1995-04-27 ltboxes.dtx v1.0s
General: Allow initial patch level	\backslash @frameb@x: Move \backslash leavevmode
0	for graphics/1512 385
1995-04-21 ltoutenc.dtx v1.7h	\backslash @iframbox: Move \backslash leavevmode
General: Added \backslash null \backslash k	for graphics/1512 385
latex/1274	\backslash @iirsbox: Move \backslash leavevmode for
1995-04-22 ltfiles.dtx v1.0p	graphics/1512 390
\backslash includeonly: Allow blanks in	\backslash @irsbox: Move \backslash leavevmode for
argument	graphics/1512 390
1995-04-22 ltmiscen.dtx v1.0x	\backslash fbox: Move \backslash leavevmode for
General: Removed extra def of	graphics/1512 384
\backslash @gobble	\backslash raisebox: Move \backslash leavevmode for
1995-04-23 ltsect.dtx v1.0j	graphics/1512 390
\backslash addcontentsline: Use	1995-04-27 ltfiles.dtx v1.0r
\backslash contentsline internally. . .	\backslash document: Added \backslash global to
1995-04-24 ltbibl.dtx v1.1e	support groups in hook 91
\backslash @citex: Add \backslash mbox to undefined	1995-04-27 ltmiscen.dtx v1.0y
case: latex/1239.	\backslash enddocument: \backslash @checkend moved
1995-04-24 ltbibl.dtx v1.1f	after hook 337
\backslash bibcite: Make \backslash @onlypreamble	1995-04-27 ltplain.dtx v1.1i
/1388.	General: Move \backslash hang and
1995-04-24 ltcntrl.dtx v1.0d	\backslash textindent to latex209.def . . . 28
\backslash @for: Dont expand second	1995-04-29 ltcntrl.dtx v1.0e
argument with \backslash edef: /1317	General: Moved init of \backslash protect
(DPC)	to ltdefns.dtx 59
1995-04-24 ltoutput.dtx v1.1j	Removed unused defs for
\backslash fl@tracemessage: Do not add to	\backslash @setprotect and
kernel unless ‘trace’ specified	\backslash @resetprotect 59
1995-04-24 ltoutput.dtx v1.1l	1995-04-29 ltdefns.dtx v1.2j
\backslash @beginvibox: Add \backslash vbox	\backslash protect: Init \backslash protect here . . . 46
latex/1392	1995-04-29 ltpar.dtx v1.1b
\backslash @writesetup: Reset \backslash	General: (TO) Comments
latex/1451 (DPC)	clean-up. 69
1995-04-24 ltpage.dtx v1.0f	1995-05-02 ltsect.dtx v1.0l
\backslash fussy: reset \backslash emergencystretch	\backslash @dottedtocline: Don’t reset to
latex/1344	\backslash rmfamily 451

1995-05-03 ltsect.dtx v1.0m	Check for format older than 1 year	33
General: TO: Promoted documentation to doc.sty standard		441
1995-05-06 ltsect.dtx 1.0n		
\@secCNTformat: Use \quad instead of \hskip		447
\@sect: Added \relax after \@secCNTformat just in case		445
1995-05-07 ltboxes.dtx v1.0t		
General: Use \hb@xt@		380
1995-05-07 ltdefs.dtx v1.2k		
\hb@xt@: Macro added		36
1995-05-07 ltmath.dtx v1.0r		
General: Use \hb@xt@		351
1995-05-07 ltoutput.dtx v1.1m		
General: Use \hb@xt@		481
1995-05-07 ltpictur.dtx v1.0g		
General: Use \hb@xt@		414
1995-05-07 ltplain.dtx v1.1j		
General: Use \hb@xt@		15
1995-05-07 ltsect.dtx v1.0o		
General: Use \hb@xt@		441
1995-05-07 lttab.dtx v1.0l		
General: Use \hb@xt@		392
1995-05-08 ltbibl.dtx v1.1g		
\@citex: Use \@firstofone		475
\@bitem: Removed unnecessary braces		475
\@nocite: Use \@firstofone		476
1995-05-08 ltdefs.dtx v1.2k		
\typein: Use \@firstofone		37
1995-05-08 ltdefs.dtx v1.2l		
\typein: Remove unnecessary braces		37
Replace \def by \let		37
1995-05-08 ltfssstrc.dtx v2.3n		
\ifnot@nil: Use \@firstofone		211
1995-05-11 fontdef.dtx v2.2j		
General: Updates to some plain macros		268
1995-05-12 ltclass.dtx v1.0n		
\DeclareOption*: Use \toks@ to remove need to double hash /1557		562
1995-05-12 ltfloat.dtx v1.1h		
\@footnotemark: Add \nobreak to allow hyphenation. latex/1605		471
1995-05-12 ltpictur.dtx v1.0h		
\pictur@: Macro added for latex/1355		415
1995-05-12 ltvers.dtx v1.0e		
General: Add autoload docstrip guards		33
1995-05-13 ltfssstrc.dtx v2.3o		
General: Use single hash mark in \DeclareOption		200
1995-05-16 ltfloat.dtx v1.1i		
\@makefnmark: Now use \textsuperscript.		469
\textsuperscript: Command added./pr1503		470
\thefootnote: Streamlined parts of code.		469
1995-05-17 ltboxes.dtx v1.0u		
\@irsbox: Removed surplus braces		390
1995-05-17 ltclass.dtx v1.0o		
\g@addto@macro: Make long for latex/1522		570
1995-05-17 ltlsts.dtx v1.0g		
\@item: Removed surplus braces		376
\@nitem: Removed surplus braces		376
enumerate: Use \thr@@ and remove surplus braces		377
itemize: Use \thr@@		378
1995-05-18 ltfloat.dtx v1.1j		
\@makefnmark: Added \normalfont.		469
\thempfootnote: Added \itshape.		469
1995-05-19 ltpictur.dtx v1.1a		
General: Support autoloading feature		414
1995-05-20 ltcounts.dtx v1.1b		
\@definecounter: Streamlined code		147
\@fnsymbol: Allowing both text and math		150
\fnsymbol: Streamlined code		149
1995-05-20 ltcounts.dtx v1.1c		
\@definecounter: And do it right		147
1995-05-20 ltfloat.dtx v1.1k		
\@makefnmark: Moved \normalfont back and use \@textsuperscript		469
Moved \normalfont to \textsuperscript		469
\textsuperscript: Use \normalfont.		470
1995-05-21 ltfssdcl.dtx v2.1t		
\DeclareMathRadical: Allow for undefined cs names		248

1995-05-21 ltlists.dtx v1.0f	1995/05/20 v2.2d	222
General: Moved to doc.sty	1995-05-24 ltssdcl.dtx v3.0a	
standard	General: (DPC) Make file from	
364	previous file, latint.dtx	
1995-05-21 ltmath.dtx v1.0r	1995/05/21 v2.1t	226
\@sqrt: Use \sqrtsign	1995-05-24 ltssini.dtx v3.0a	
359	General: (DPC) Make file from	
General: Remove \mathhexbox	previous file, lfont.dtx	
from this file	1995/05/23 v2.2e	252
356	\cal: (DPC) Remove definition .	267
Update some plain macros . . .	\mit: (DPC) Remove definition .	267
351	1995-05-24 ltssstrc.dtx v3.0a	
\leftteqn: Use \rlap	General: (DPC) Make file from	
360	previous file, tracefnt	
\right: Use \sqrtsign instead of	1995/05/16 v2.3o	199
\sqrt	1995-05-24 ltssstrc.dtx v3.0b	
353	General: (DPC) Fix	
1995-05-21 ltoutenc.dtx v1.7h	\ProvidesFile usage	199
\@inmathwarn: Added several	1995-05-25 ltclass.dtx v1.0p	
\@onlypreamble	\endfilecontents: Delete	
106	\filecontents after preamble	571
1995-05-21 ltoutenc.dtx v1.7j	1995-05-25 ltfiles.dtx v1.0s	
General: Updated some plain	\document: Added check for	
macros	\topskip zero	91
118	1995-05-25 ltfiles.dtx v1.0t	
1995-05-21 ltplain.dtx v1.1j	\@iffilenamepath: (CAR) added	
General: Moved some code to	\long	97
other files	\document: Corrected typo	91
15	\IfFileExists@: (CAR) added	
1995-05-22 ltplain.dtx v1.1k	\long	97
General: Definitions of \footins	\InputIfFileExists: (CAR)	
and \footnoterule moved to	added \long	98
ltfloat.	\nofiles: (CAR) added \long . .	93
30	\protected@write: (CAR) added	
1995-05-22 lttab.dtx v1.1a	\long	94
General: Support autoloading	1995-05-25 ltfloat.dtx v1.1m	
feature	\@savemarbox: (CAR) Resettings	
392	moved to hook	465
1995-05-23 ltssini.dtx v2.2e	\@xfloat: (CAR) Resettings	
\newfont: Font assignment made	moved to hook	458
local again.	1995-05-25 ltlists.dtx v1.0i	
264	\endtrivlist: Macros moved from	
1995-05-24 ltdefns.dtx v1.1l	ltspace.dtx	373
\newif: (DPC) New	1995-05-25 ltmath.dtx v1.3c classes	
implementation	\@eqnnum: replace	
42	\reset@font\rmfamily with	
1995-05-24 ltdefns.dtx v1.2m	\normalfont (PR 1578) . . .	361
\typein: (DPC) New	1995-05-25 ltspaced.dtx v1.2f	
implementation	\@vbsphack: (CAR) not used so	
37	‘removed’.	80
1995-05-24 ltfloat.dtx v1.1l	\@vspacer: (CAR) \@restorepar	
\@textsuperscript: Command	added to avoid possible infinite	
added.	tail recursion caused by a typo	
470	in the argument.	83
General: Moved definition of		
\footins and \footnoterule		
from ltplain.		
469		
\textsuperscript: Use		
\@textsuperscript		
470		
1995-05-24 ltssbas.dtx v3.0a		
General: (DPC) Make file from		
previous file, fam.dtx		
1995/05/20 v2.2d		153, 176
\mathgroup: (DPC) No need to		
redefine \newfam as not		
outer		153, 176
1995-05-24 ltsscmp.dtx v3.0a		
General: (DPC) Make file from		
previous file, fam.dtx		

(CAR) macros modified to be more efficient	83	1995-06-13 ltfinal.dtx v1.0t	General: Add patch level string more carefully	625
General: Macros moved to ltlists.dtx	71	1995-06-13 ltpictur.dtx v1.1b	Call <code>\errorstopmode</code>	627
1995-05-26 ltdefns.dtx v1.2n		General: Use <code>\ProvidesFile</code> in autoload	414	
<code>\@gobblefour</code> : (CAR) Added <code>\longs</code>	44	1995-06-14 lttab.dtx v1.1b	General: Use <code>\ProvidesFile</code> in autoload	392
1995-05-26 ltmath.dtx v1.0s		1995-06-15 ltssbas.dtx v3.0c	General: (DPC) minor documentation changes	153, 176
<code>\@eqnnum</code> : Removed <code>\rmfamily</code> (PR 1578), replaced <code>\reset@font</code> with <code>\normalfont</code>	358	1995-06-15 ltsscmp.dtx v3.0b	General: (DPC) minor documentation edits	222
1995-05-26 ltpage.dtx v1.0g		1995-06-15 ltssdcl.dtx v3.0b	General: (DPC) minor documentation changes	226
<code>\ps@plain</code> : removed <code>\rmfamily</code> (PR 1578)	479	1995-06-19 ltbibl.dtx v1.1h	<code>\biblecite</code> : Call <code>\@newl@bel</code> so repeated keys produce better warning.	475
1995-05-27 ltssbas.dtx v3.0b		1995-06-19 ltclass.dtx v1.0q	<code>\documentclass</code> : Dont redefine <code>\usepackage</code> in compat mode for /1634	565
<code>\mathgroup</code> : (FMi) But a need to define <code>\new@mathgroup</code>	153, 176	1995-06-19 ltxref.dtx v1.1e	<code>\newlabel</code> : Use <code>\@newl@bel</code> to share code with <code>\biblecite</code>	334
1995-06-05 fontdef.dtx v2.2k		1995-06-28 ltssini.dtx v3.0b	General: (DPC) Fix documentation typos	252
General: Moved math commands from ltoutenc.dtx.	286	1995-06-28 ltmath.dtx v1.0t	General: minor doc edits	351
1995-06-05 ltfinal.dtx v1.0r		1995-07-02 ltplain.dtx v1.1n	General: Removed surplus ‘by’ and ‘=’ in various places	15
General: Added <code>\MakeUppercase</code> and <code>\MakeLowercase</code>	613	<code>\offinterlineskip</code> : Replaced 1000 by <code>\@m</code>	27	
1995-06-05 ltoutenc.dtx v1.7k		<code>\showoutput</code> : Use <code>\showoverfull</code> to save space	30	
<code>\@inmathwarn</code> : Removed <code>\protected@cmd</code> and replaced with explicit <code>\noexpand</code>	106	<code>\tracingall</code> : Use <code>\showoutput</code> to save space	30	
General: Allowed <code>\ProvideTextCommandDefault</code> after the preamble.	108	1995-07-03 ltdefns.dtx v1.2o	<code>\set@typeset@protect</code> : Use <code>\@typeset@protect</code> for init	46
Commented out <code>\textless</code> and <code>\textgreater</code>	115	1995-07-03 ltfontcmd.dtx v3.3s	<code>\t@st@ic</code> : Use clean interface for jump	298
Moved math commands to fontdef.dtx.	118	1995-07-05 ltfontcmd.dtx v3.3s	<code>\t@st@ic</code> : Renamed from <code>\test@next</code>	297
Save some tokens in <code>\textvisiblespace</code> and <code>\textunderscore</code>	115			
1995-06-06 ltfinal.dtx v1.0s				
General: Made <code>\MakeUppercase</code> and <code>\MakeLowercase</code> brace their argument.	613			
1995-06-09 ltoutenc.dtx v1.7l				
<code>\DeclareTextComposite</code> : Rewrote <code>\DeclareTextComposite</code> to define the composite as a no-argument command rather than a two-argument command.	109			
1995-06-11 ltspc.dtx v1.2g				
<code>\restorecr</code> : (CAR) <code>\relax</code> added to stop silent eating of *.	86			

1995-07-05 ltspace.dtx v1.2h		in new \enddocument	475
\@gnewline: Use \break	76	1995-07-14 ltxref.dtx v1.1g	
\@no@pgbk: Macro replaces \@pgbk		\newlabel: Remove	
and \@nopgbk	75	\@onlypreamble so still defined	
\nopagebreak: Reimplemented		in new \enddocument	334
both using \@no@pgbk	74	1995-07-19 ltfssini.dtx v3.0d	
1995-07-09 ltctrl.dtx v1.0f		General: (DPC) TeX2 support . .	266
\@iforloop: Reimplemented using		1995-07-20 ltboxes.dtx v1.0v	
Kabelschacht method	59	\@isavebox: Use \sbox	383
\@iwhiledim: Reimplemented		\@isavepicbox: Use \sbox	383
using Kabelschacht method . .	57	1995-07-21 ltoutput.dtx v1.1o	
\@iwhilenum: Reimplemented		\@writesetup: Command added .	508
using Kabelschacht method . .	57	New, experimental, versions:	
\@iwhilesw: Reimplemented using		need in-lining	508
Kabelschacht method	57	1995-08-09 ltmath.dtx v1.0u	
\@tfor: Reimplemented using		General: Added code for class	
Kabelschacht method	59	options leqno and fleqn	360
1995-07-09 ltlists.dtx v1.0j		1995-08-11 ltlength.dtx v1.1b	
enumerate: Use \expandafter . .	378	General: Doc typos fixed for	
itemize: Use \expandafter	378	latex/753	152
1995-07-12 ltpictur.dtx v1.1d		1995-08-16 ltctrl.dtx v1.0g	
General: allow 2e commands in 209		\@break@tfor: Made long	59
mode. latex/1737	414	\@forloop: Made defs long	58
1995-07-13 ltdefs.dtx v1.0p		\@fornoop: Made defs long	58
General: Updates to		\@iforloop: Made defs long	59
documentation	36	\@iwhiledim: Made defs long . . .	57
1995-07-13 ltfiles.dtx v1.0u		Removed \@whilenoop	57
General: Updates to docu	88	\@iwhilenum: Made defs long . . .	57
1995-07-13 ltfssbas.dtx v3.0d		Removed \@whilenoop	57
\@defaultsubs: macro		\@iwhilesw: Removed	
added	170, 193	\@whileswnoop	57
\@defaultsubs: macro added	170, 193	\@tfor: Made defs long	59
General: minor documentation		1995-08-16 ltfiles.dtx v1.0v	
changes	153, 176	\document: set \@maxdepth	91
\@wrong@fontshape: Change a		set \do globally	92
macro not a switch to flag		set \topskip globally	91
default font substitutions	169, 192	1995-08-24 ltfssbas.dtx v3.0f	
1995-07-13 ltmiscen.dtx v1.0z		General: Added autoloader	
\@centercr: Use \nobreak	344	code	153, 176
\@writefile: Added missing		1995-08-24 ltfssstrc.dtx v3.0c	
percent and use \relax in the		General: Macro	
THEN case	340	\gobble@font@spec removed	211
\@xobeysp: Use \nobreak	346	\@tryis@simple:	218
General: Improve Documentation	336	1995-08-25 ltoutput.dtx v1.1p	
\enddocument: Set \@setckpt to		General: Support autoloading	
\@gobbletwo instead of		feature (FMi).	481
defining it by hand	337	1995-09-01 lterror.dtx v1.2i	
Shorten redefinition of \bibtex		General: Add autoloader support .	60
and \newlabel	338	1995-09-01 ltplain.dtx v1.1m	
Use \@defaultsubs instead of		\empty: Use \let to save space .	27
switch	338	\I: Use \let to save space	26
1995-07-14 ltbibl.dtx v1.1i		1995-09-14 ltplain.dtx v1.1o	
\bibtex: Remove		General: Moved \multispan to	
\@onlypreamble so still defined		lttab.dtx	15

1995-09-14 lttab.dtx v1.1c	1995-10-10 ltthm.dtx v1.0f
\cline: (DPC) New	General: Make \newtheorem ‘only
implementation 412	preamble’ 437
1995-09-15 ltssini.dtx v3.0e	1995-10-11 ltoutput.dtx v1.1r
General: (DPC) Modify TeX2	\clearpage: Added a check so that
message 266	it does not lose the argument
1995-09-19 ltmiscen.dtx v1.1a	of \twocolumn[...] 494
\verb: Put \enoligs after	1995-10-16 ltbibl.dtx v1.1j
\verbatim@font where it	\cite: (DPC) Make robust 475
belongs. 349	1995-10-16 ltboxes.dtx v1.0w
1995-10-01 ltfiles.dtx LaTeX2e	General: Clarify makebox
\@addtofilelist: Macro added . 100	description 380
1995-10-02 ltdefs.dtx v1.2q	1995-10-16 ltdefs.dtx v1.2u
\ifnch: Use \let@token for	\ifstar: (DPC) New
internal/924, save \reserved@e 50	implementation, for /1910 . . . 50
\ifnextchar: Use \let@token 50	\newcommand: (DPC) Use
\protected@testopt: Macro	\testopt /1911 39
added 40	\newenvironment: (DPC) Use
\testopt: Macro added 39	\testopt /1911 41
\@xargdef: New implementation,	\typein: (DPC) Use \testopt
using \testopt 39	/1911 37
1995-10-03 fontdef.dtx v2.2l	1995-10-16 ltssini.dtx v3.0f
General: \@@sqrt from patch file	\p@reset@font: Added \relax
for /1701 268	after \usefont, as the latter
1995-10-03 ltdefs.dtx v1.2r	eats up spaces. 265
\typein: Add missing \typein	1995-10-16 ltmath.dtx v1.0y
for /1710 (from patch file) . . . 38	\yeqncr: (DPC) Use \testopt
1995-10-03 ltpictur.dtx v1.1e	/1911 359
General: New autoloader code . . . 414	\sqr: (DPC) Make robust /1808 359
1995-10-04 ltssbas.dtx v3.0g	1995-10-16 ltspace.dtx v1.2j
General: Modify autoloader	\nolinebreak: (DPC) Use
code 153, 176	\testopt /1911 74
1995-10-04 ltssstrc.dtx v3.0d	\nopagebreak: (DPC) Use
General: (DPC) Modify autoloader	\testopt /1911 74
code 199	1995-10-16 ltthm.dtx v1.0g
1995-10-04 lttab.dtx v1.1d	General: Revert to previous
General: Modify autoloader support 392	\newtheorem behaviour 437
1995-10-06 ltfiles.dtx v1.0w	1995-10-17 ltclass.dtx v1.0r
\@missingfileerror: Autoloader	\@providesfile: Delay definition
error 99	of \ProvidesFile till ltfinal 562
1995-10-09 lterror.dtx v1.2j	\ProcessOptions*: Reset
General: Modify autoloader support 60	\CurrentOption for
1995-10-09 ltoutenc.dtx v1.7m	graphics/1873 564
\@inmathwarn: Autoloader error . . 107	1995-10-17 ltdirchk.dtx v1.0l
1995-10-10 ltssbas.dtx v3.0h	General: Modify initex version of
\showhyphens: Use \normalfont	\ProvidesFile 4
and make colour safe, and	1995-10-17 ltfinal.dtx v1.0v
autoloadable 173, 196	\@providesfile: reset macro . . . 627
1995-10-10 ltssdcl.dtx v3.0c	\reserved@b: reset here after the
\non@alpherr: (DPC) autoloader	\input above 626
error message 230	1995-10-17 ltplain.dtx v1.1s
1995-10-10 ltplain.dtx v1.1r	\reject: Move \supereject to
General: Autoloader tracing code . 15	compat file 28

1995-10-17 lttab.dtx v1.1e		1995-10-24 ltmiscen.dtx v1.1b	
\@cline: (DPC) Use \@multicnt	413	\enddocument: Changed logic for	
\@multispan: (DPC) Macro		producing warning messages	
added.	413	and removed switch	338
1995-10-19 ltfinal.dtx v1.0w		Use \@refundefined instead of	
\@filelist: Move after		switch	338
\reserved@a setting:-)	627	1995-10-24 ltxref.dtx v1.1h	
1995-10-20 ltbibl.dtx v1.1k		\@multiplelabels: Switch for	
\@citex: Removed refundefined		multiplelabels removed	334
flag	475	\@newl@bel: Switch for	
\nocite: Removed refundefined		multiplelabels replaced by	
flag	476	inline code	334
1995-10-20 ltclass.dtx v1.0s		\@refundefined: Switch for	
\@begindocumenthook: Make		refundefined replaced	333
setting conditional, for		\@setref: Switch for refundefined	
autoload version	570	renamed	334
1995-10-20 ltfsbas.dtx v3.0i		\if@multiplelabels: Macro	
General: (DPC) Modify autoload		removed	334
code, change \undefined	153, 176	1995-10-25 ltalloc.dtx v1.1b	
1995-10-20 ltfsstrc.dtx v3.0e		General: General doc	
General: (DPC) Modify autoload		improvements	54
code	199	1995-10-25 ltfloat.dtx v1.1n	
1995-10-22 ltfsbas.dtx v3.0j		\@endfloatbox: (CAR) macro	
General: (RmS) New size function		added: to unify code for double	
macro \genb@sfcnt needs to be		and single versions	462
disabled at \document. . .	153, 176	\end@dblfloat: (CAR) unify code	
1995-10-22 ltfsstrc.dtx v3.0f		for double and single versions	461
General: Added 'genb' and 'sgenb'		\end@float: (CAR) unify code for	
size functions to support new		double and single versions . .	460
DC font naming scheme. . .	199	1995-10-25 ltidxglo.dtx v1.1d	
1995-10-23 lttab.dtx v1.1f		General: Doc cleanup	472
\@settab: (CAR)Ensure that		1995-10-25 ltsect.dtx v1.0q	
\@hightab increases by at most		\subparagraphmark: Use \let not	
one	399	\def to save space.	448
\@startline: (CAR)Ensure that		1995-10-27 ltpictur.dtx v1.1f	
\@nxttabmar is never larger		General: Move initialisation to	
than \@hightab	397	kernel from autoload file . . .	433
\@poptabs: (CAR)Ensure that		1995-10-31 ltboxes.dtx v1.0x	
\@curtab is never larger than		\@finalstrut: Add \nobreak in	
\@hightab	400	horiz mode to allow	
\@tabbing: (CAR)Make \@hightab		hyphenation. internal/1931 . .	391
consistently a local variable .	398	1995-11-01 fontdef.dtx v2.2m	
1995-10-24 ltfiles.dtx v1.1a		General: add \nfss@catcodes for	
\document: Removed		internal/1932	271
multiplelabels switch	91	1995-11-01 ltdirchk.dtx v1.0n	
Removed refundefined switch .	91	General: Initialise	
1995-10-24 ltfsbas.dtx v3.0k		\@addtofilelist to \@gobble . .	4
\@defaultsubs: macro		1995-11-01 ltfinal.dtx v1.0x	
removed	170, 193	General: (DPC) Switch meaning of	
\wrong@fontshape: Make this code		\@addtofilelist for cfg files	618
inline since it happens only		1995-11-01 ltfsbas.dtx v3.0m	
here	169, 192	\DeclareFontShape: (DPC) Test	
		for \relax not \undefined,	
		internal/1933	154, 177

1995-11-01 ltffssini.dtx v3.0g	1995-11-17 ltoutenc.dtx v1.7p
General: (DPC) Switch meaning of \@addtofilelist for cfg files 266	\UseTextSymbol: Support \@wrong@font@char latex/1676 111
1995-11-02 ltffssbas.dtx v3.0n	1995-11-18 ltoutenc.dtx v1.7q
\wrong@fontshape: (DPC) Remove extra space with \string for latex/1676 . 169, 192	\UseTextSymbol: Modify message slightly 111
1995-11-02 ltoutenc.dtx v1.7n	1995-11-21 fontdef.dtx v2.2n
General: Changed internal name \@a to \@tabacckludge to protect against redefinition by malicious users. 114	General: Incorporate changed figures, as in plain.tex 284
1995-11-07 ltlists.dtx v1.0k	1995-11-27 ltffssbas.dtx v3.0n
\@doendpe: Enclosed \setbox0 assignment by a group so that it leaves the contents of box 0 intact. 374	\nfss@catcodes: Reset hash, for definitions in fd files ... 165, 188
1995-11-07 ltoutenc.dtx v1.7o	1995-11-28 ltfloat.dtx v1.1n
General: Added \leavevmode at start of \c, otherwise the output routine might be invoked within the macro. .. 118	General: documentation fixes ... 453
Changed \char32 to \@xxxii (two tokens less). 119	1995-11-28 ltfsstrc.dtx v3.0g
Replaced octal number 27 by decimal number 23 to protect against the quote character being active. 119	General: documentation fixes ... 199
Replaced some 0's by \z@ (faster). 119	1995-11-28 ltoutenc.dtx v1.7r
1995-11-10 ltoutput.dtx v1.1s	General: Added math mode checks to text commands. 106
\@shipoutsetup: Command removed 508	doc fixes 102
\@writesetup: Command removed 508	Renamed \@changed@x@err to \TextSymbolUnavailable. .. 106
In-lined 508	1995-11-29 ltoutenc.dtx v1.7t
1995-11-14 ltclass.dtx v1.0t	General: Added \textasciicircum, \textasciitilde, \textbackslash, \textbar, \textgreater and \textless. 121
\@unprocessedoptions: Allow empty option 571	Added \textasciicircum, \textasciitilde, \textregistered and \texttrademark. 115
\@loadwithoptions: macro added 565	Added \textbackslash and \textbar. 115, 125
\LoadClassWithOptions: macro added 566	Added \textless and \textgreater. 115, 126
\RequirePackageWithOptions: macro added 566	1995-12-01 ltoutenc.dtx v1.7u
1995-11-17 ltffssbas.dtx v3.0m	General: Made \SS a Default, rather than having the default point to the OT1 definition. 115
\@wrong@font@char: (DPC) Macro added. latex/1676 170, 193	1995-12-04 ltspac.dtx v1.2k
\define@newfont: Redefine \typeout latex/1676 ... 163, 186	\nobreakspace: (Macro added .. 84
\wrong@fontshape: Support \@wrong@font@char latex/1676 169, 192	1995-12-04 ltspac.dtx v1.2l
	\@xobeysp: (braces added to definition of tilde 84
	1995-12-04 preload.dtx v2.4e
	General: Ulrik Vieth. added 12pt OMS and OML preloads /1989 290
	1995-12-05 ltdefns.dtx 1.2w
	\@unexpandable@protect: Removed

<code>\unexpandable@noexpand</code> as never used. internal/1733 . . .	45	1996-04-22 lttab.dtx v1.1g	
1995-12-05 ltfiles.dtx v1.1c		<code>\@tabclassz:</code> (DPC) Extra <code>\hskip</code> keeps <code>tabcolsep</code> in empty columns internal/2122	411
<code>\document:</code> <code>\ignorespaces</code> added for latex/1933	92	1996-04-23 ltcounts.dtx v1.1d	
1995-12-05 ltfloat.dtx v1.1n		General: Documentation improvements	145
<code>\@textsuperscript:</code> Use <code>\ensuremath</code> for latex/1984.	470	1996-04-24 ltfiles.dtx v1.1e	
1995-12-05 ltoutenc.dtx v1.7v		<code>\document:</code> (DPC) Reset <code>\AtBeginDocument</code> eg for latex/1297	91
<code>\@inmathwarn:</code> Changed <code>\TextSymbolUnavailable</code> text	107	1996-05-08 ltfssstrc.dtx v3.0h	
1995-12-06 ltfssbas.dtx v3.00		<code>\math@egroup:</code> Use <code>\bgroup</code> instead of <code>\begingroup</code> to match a kernel change made in 1994!!	209
<code>\nfss@catcodes:</code> Reset hat, for typeouts etc in fd files	165, 188	1996-05-09 ltfntcmd.dtx v3.3t	
1995-12-07 ltbibl.dtx v1.1l		<code>\check@icr:</code> Default definitions added	296
<code>\@citex:</code> Restored name of <code>\G@refundefinedtrue</code>	475	1996-05-17 fontdef.dtx v2.2o	
1995-12-07 ltfloat.dtx v1.1m		General: <code>\@sqr</code> removed, at last	268, 284
<code>\@textsuperscript:</code> Move <code>\m@th</code> out of the <code>\ensuremath</code> for latex/1984.	470	1996-05-17 ltfiles.dtx v1.1f	
1995-12-07 ltxref.dtx v1.1i		<code>\nofiles:</code> added <code>\write</code> to <code>\protected@write</code> for latex/2146	93
<code>\@setref:</code> Switch for <code>refundefined</code> restored	334	1996-05-18 ltoutenc.dtx v1.7x	
<code>\G@refundefinedtrue:</code> Renamed (back) from <code>\G@refundefined</code>	333	General: Produce error if encoding not found. pr/2054	142
1995-12-11 ltoutenc.dtx v1.7w		1996-05-21 ltoutenc.dtx v1.7y	
1995-12-13 ltdefns.dtx 1.2x		General: Corrected error message (CAR)	142
<code>\-:</code> Documentation changed.	51	1996-05-21 ltsect.dtx v1.0s	
1996-01-10 ltfiles.dtx v1.1d		<code>\@sect:</code> (DPC) Added extra braces for internal/2148	445
<code>\@iffilenamepath:</code> Change argument handling to not require doubled hash. latex/2024	97	(DPC) Moved brace to allow commands like <code>\MakeUppercase</code> in 6th argument. Changed <code>\par</code> to <code>\endgraf</code> to allow non-long commands. internal/2148	445
1996-01-20 ltidxglo.dtx v1.1e		<code>\@ssect:</code> (DPC) Added extra braces for internal/2148	448
<code>\makeglossary:</code> Make no-op after use pr/2048	473	(DPC) Moved brace to allow commands like <code>\MakeUppercase</code> in 4th argument. Changed <code>\par</code> to <code>\endgraf</code> to allow non-long commands. internal/2148	447
<code>\makeindex:</code> Make no-op after use pr/2048	473	1996-05-23 ltoutenc.dtx v1.7z	
1996-01-20 ltspc.dtx v1.2m		<code>\@strip@args:</code> <code>\expandafter</code> added to match other changes for latex/2133	111
<code>\vspace:</code> Made robust	83		
1996-03-25 ltmath.dtx v1.1a			
<code>\@ensuredmath:</code> Macro added for amslatex/2104	360		
<code>\ensuremath:</code> Reimplement for amslatex/2104	360		
1996-04-18 ltpage.dtx v1.0i			
General: Improve documentation	478		
1996-04-22 ltmiscen.dtx v1.1c			
General: Improve Documentation	336		
1996-04-22 ltspc.dtx v1.2n			
General: Documentation Improvements	71		

<code>\add@accent</code> : macro added. <code>latex/2133</code> 108	1996-06-22 <code>ltspace.dtx</code> v1.2o General: Documentation of problems added 71
<code>\DeclareTextAccent</code> : Reimplemented using <code>\add@accent</code> to save space <code>latex/2133</code> 108	1996-07-10 <code>ltfinal.dtx</code> v1.0y <code>\toks</code> : Free up memory from scratch registers /2213 626
<code>\DeclareTextCompositeCommand</code> : Modified to cope with new <code>\add@accent</code> command: required removal of check for one argument-command ... 109	1996-07-19 <code>ltoutenc.dtx</code> v1.8a <code>\@strip@args</code> : Use char 0 not @ as carrier for <code>\lowercase</code> /2197 111
1996-05-24 <code>ltoutput.dtx</code> v1.1t <code>\@specialoutput</code> : Check that <code>\@colroom</code> is less than <code>\vsize</code> , indicating that a float has been added 499	1996-07-26 <code>ltboxes.dtx</code> v1.0z <code>\if@minipage</code> : put <code>\global</code> into definition 388
Cut-off point changed to 1.5 <code>\baselineskip</code> 499	1996-07-26 <code>ltclass.dtx</code> v1.0u <code>\@classoptionslist</code> : made only preamble 557
<code>\@topnewpage</code> : Cut-off point changed to 2.5 <code>\baselineskip</code> 498	<code>\@unusedoptionlist</code> : made only preamble 558
1996-05-25 <code>ltoutput.dtx</code> v1.1u <code>\@specialoutput</code> : Correct the above check 499	1996-07-26 <code>ltdefns.dtx</code> v1.2y <code>\@reargdef</code> : third arg picked up by <code>\@yargdef</code> 41
1996-06-03 <code>ltmiscen.dtx</code> v1.1d <code>\@verbatim</code> : Exchanged the following two code lines so that <code>\dospecials</code> cannot reset the category code of characters handled by <code>\@noligs</code> 347	<code>\renew@command</code> : use <code>\noexpand</code> instead of <code>\string</code> 41
General: Move setting of verbatim font and <code>\@noligs</code> 336	use <code>\relax</code> in place of empty arg 41
<code>\verb</code> : Put setting of verbatim font after <code>\dospecials</code> so that <code>\dospecials</code> cannot reset the category code of characters handled by <code>\@noligs</code> 349	<code>\renew@environment</code> : use <code>\relax</code> in place of empty arg 42
1996-06-10 <code>ltboxes.dtx</code> v1.0y <code>\@parboxto</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code> 386	1996-07-26 <code>lfloat.dtx</code> v1.1n <code>\@endfloatbox</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code> 462
1996-06-10 <code>ltsect.dtx</code> v1.0t <code>\@sect</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code> 445	<code>\@savemarbox</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code> 465
<code>\@ssect</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code> 447	<code>\@setminipage</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code> 460
1996-06-13 <code>ltdirchk.dtx</code> v1.0r General: documentation improvements mainly from internal/2174 1	<code>\@setnobreak</code> : remove unnecessary <code>\global</code> before <code>\@nobreak...</code> 460
1996-06-14 <code>lftab.dtx</code> v1.1h <code>\@tabclassz</code> : (DPC) Change both <code>\z@skip</code> to 1sp for <code>latex/2160</code> 411	1996-07-26 <code>ltfssbas.dtx</code> v3.0p <code>\@DeclareMathSizes</code> : use faster <code>\if</code> test 158, 181
	<code>\nfss@catcodes</code> : omit <code>\relax</code> as not needed 165, 188
	1996-07-26 <code>ltfssdcl.dtx</code> v3.0e <code>\init@restore@version</code> : Removed <code>\ifrestore@version</code> switch and replaced by <code>\init@restore@version</code> ... 230
	1996-07-26 <code>ltfssstrc.dtx</code> v3.0i <code>\init@restore@glb@settings</code> : macro added replacing <code>\ifinmath</code> switch 208
	1996-07-26 <code>ltlists.dtx</code> v1.0l <code>\@item</code> : Remove unnecessary <code>\global</code> before

<code>\@minipage...</code>	375	1996-08-02 ltsect.dtx v1.0v	
Remove unnecessary <code>\global</code>		<code>\@afterheading</code> : Removed	
before <code>\@nobreak...</code>	376	<code>\global</code> before <code>\@nobreak...</code>	448
1996-07-26 ltmath.dtx v1.1b		1996-08-02 ltspc.dtx v1.2s	
General: Removed <code>\global</code> before		<code>\@Esphack</code> : Remove <code>\global</code>	
<code>\@ignoretrue</code> in various		before <code>\@ignore...</code>	79
places.	351	1996-08-25 ltssbas.dtx v3.0r	
1996-07-26 ltmiscen.dtx v1.1e		<code>\nfss@catcodes</code> : Reset the acute,	
<code>\@ignorefalse</code> : put <code>\global</code> into		grave and double quote chars	
definition	337	as well	165, 188
<code>\begin</code> : remove <code>\global</code> before		1996-09-21 ltoutput.dtx v1.1w	
<code>\@ignore...</code>	341	<code>\@writesetup</code> : Added	
<code>\end</code> : remove <code>\global</code> before		<code>\@parboxrestore</code> and made	
<code>\@ignore...</code>	343	consequent deletions: wait for	
<code>\ignorespacesafterend</code> : user level		the howls of protest	508
macro added	337	1996-09-25 ltdirchk.dtx v1.0t	
1996-07-26 ltoutput.dtx v1.1v		General: Move ltxcheck to separate	
<code>\@testfp</code> : remove <code>\global</code> before		file	14
<code>\@test...</code>	541	1996-09-28 ltmiscen.dtx v1.1f	
<code>\@xtryfc</code> : remove <code>\global</code> before		<code>\@xobeysp</code> : Moved to ltspc.dtx	346
<code>\@test...</code>	516	1996-09-28 ltspc.dtx v1.2t	
<code>\@ztryfc</code> : remove <code>\global</code> before		<code>\@xobeysp</code> : Moved from	
<code>\@test...</code>	517	ltmiscen.dtx and redefined to	
General: put <code>\global</code> into		use <code>\nobreakspace</code>	84
definition	491	1996-09-29 ltfiles.dtx v1.1g	
remove <code>\global</code> before		<code>\document</code> : Added disabling of	
<code>\@test...</code>	491, 492	<code>\@nodocument</code>	92
<code>\clearpage</code> : add number of		1996-09-29 ltoutput.dtx v1.1x	
missing percents	494	<code>\newpage</code> : Checks for noskipsec	
1996-07-26 ltplain.dtx v1.1t		and inlabel added	495
<code>\sh@ft</code> : replace <code>\dimen@z@</code> by		1996-09-29 ltsect.dtx 1.0w	
<code>\dimen@</code>	29	<code>\@noskipsectrue</code> : Added	
1996-07-26 ltsect.dtx v1.0u		documentation	442
<code>\@starttoc</code> : removed <code>\global</code>		1996-09-30 ltoutput.dtx v1.1y	
before <code>\@nobreak...</code>	449	<code>\newpage</code> : Checks for noskipsec	
<code>\@xsect</code> : Removed <code>\global</code> before		and inlabel removed pending	
<code>\@nobreak...</code>	446	further tests	495
1996-07-26 ltspc.dtx v1.2p		1996-10-04 ltclass.dtx v1.0v	
<code>\if@nbreak</code> : put <code>\global</code> inside		<code>\RequirePackageWithOptions</code> :	
definition	76	Reset <code>\@unprocessedoptions</code>	
1996-07-27 ltssbas.dtx v3.0q		for /2269	566
General: <code>\if@inmath</code> switch		1996-10-05 ltfiles.dtx v1.1h	
removed	163, 186	<code>\@clubpenalty</code> : Added setting its	
1996-07-27 ltspc.dtx v1.2q		value	90
General: Further documentation of		1996-10-08 ltfontcmd.dtx v3.3u	
problems	71	<code>\DeclareTextFontCommand</code> :	
1996-07-27 ltspc.dtx v1.2r		Removed <code>\check@icr</code> when in	
General: Correct documentation of		vmode since it causes various	
problems	71	errors (see pr/2157)	294
1996-08-02 ltfloat.dtx v1.1o		1996-10-21 lttab.dtx v1.1i	
<code>\@xympar</code> : Remove <code>\global</code> before		<code>\array</code> : Use	
<code>\@ignore...</code>	466	<code>\set@typeset@protect</code>	404
		General: Moved the code	
		associated with <code>\@mkpream</code> into	

- the group provided by the box,
for robustness (latex/2183) . 404
- `\multicolumn`: Make
`\multicolumn` long
(latex/2180) 406
- `\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) 398
- 1996-10-23 ltlists.dtx v1.0m
`\@item`: `\@nobreak`... moved into
the `\everypar` and not
executed unconditionally, see
above 376
- `\kern`... changed to
`\setbox`... 375
- Added setting of `\clubpenalty`
and set `\@nobreakfalse` only
when necessary 376
- 1996-10-23 ltsect.dtx v1.0x
`\@xsect`: Replaced `\hskip`...
with `\setbox`... as used in
`\@afterheading` 446
- 1996-10-24 ltboxes.dtx v1.1a
`\@arrayparboxrestore`: Added
local settings of flags:
dangerous! 387
- `\@iiiminipage`: Use it or lose it
(`\@setminpage`): Frank will
want to lose it 388
- 1996-10-24 ltfloat.dtx v1.1p
`\@floatboxreset`: Added local
settings of flags: dangerous! . 460
- `\@marginparreset`: Added local
settings of flags: dangerous! . 466
- `\@xfloat`: Added `\@nodocument` to
trap floats in the preamble . 457
- 1996-10-24 ltoutput.dtx v1.1z
`\@addtocurcol`: Added `\nobreak`,
etc as appropriate 521, 525
- `\@specialoutput`: Added
`\nobreak` as appropriate . . . 501
- `\@topnewpage`: Added
`\@nodocument` to trap
`\twocolumn` in the preamble 496
- `\newpage`: Better checks for
`noskipsec` and `inlabel` added,
plus `nobreak` 495
- 1996-10-25 ltlists.dtx v1.0n
`\endtrivlist`: Change `\indent` to
`\leavevmode` 373
- Reset flags explicitly 373
- 1996-10-25 ltoutput.dtx v1.2a
`\newpage`: Reset all flags
explicitly 495
- 1996-10-26 ltlists.dtx v1.0o
`\endtrivlist`: Correct typo 373
- 1996-10-27 ltoutenc.dtx v1.8c
`\@strip@args`: Removed macro . 109
- General: Added `\r A` 119
- Added
`\textasteriskcentered` 115, 125
- Corrected syntax descriptions . 103
- Removed `\aa` and `\AA` 115, 119, 121
- 1996-10-28 ltplain.dtx v1.1u
General: (CAR) More doc changes 15
- `\dotfill`: Removed math mode . 30
- 1996-10-29 ltplain.dtx v1.1v
`\dotfill`: Got arithmetic correct
(CAR) 30
- 1996-10-29 ltspace.dtx v1.2u
`\gnewline`: Added macro 76
- `\@no@lnbk`: Macro replaces `\@lnbk`
and `\@nolnbk` 75
- `\@`: Corrected and rationalised code 75
- `\nolinebreak`: Reimplemented
both using `\@no@lnbk` 74
- 1996-10-31 ltfinal.dtx v1.0z
General: Added extra `\lcode`,
hoping it does no harm in T1
(pr/1969) 617, 624
- 1996-10-31 ltlists.dtx v1.0p
`\@trivlist`: Added check for
missing item in outer list . . . 372
- 1996-10-31 ltsect.dtx v1.0y
General: Corrected and tidied
documentation; removed long
lines 441
- 1996-11-03 ltplain.dtx v1.1w
`\dotfill`: Saved tokens by using
`\hb@xt@` 30
- 1996-11-04 lterror.dtx v1.2m
`\@nodocument`: Always define
`\@nodocument` in kernel, so that
it can be cleared by `\document`. 66
- 1996-11-04 ltlists.dtx v1.0q
`\@trivlist`: Moved check for
missing item: only checked
when not `inlabel` flag is false 372
- 1996-11-05 ltfiles.dtx v1.1i
`\nofiles`: Standard `\if@nobreak`
test added 93
- 1996-11-09 ltmath.dtx v1.1c
`\@ensuredmath`: Made long, as it
was before. /2104 360

- 1996-11-18 ltffssbas.dtx v3.0s
`\define@newfont:` (DPC)
 lowercase fd file names.
 internal/1044 164, 187
- 1996-11-18 ltoutenc.dtx v1.8d
 General: (DPC) lowercase external
 file names. internal/1044 . . . 142
- 1996-11-20 fontdef.dtx v2.2p
 General: lowercase fd and enc.def
 file names /1044 268
- 1996-11-20 ltvers.dtx v1.0f
 General: Check for old format
 modified /2319 33
- 1996-11-23 ltoutenc.dtx v1.8e
 General: Corrected description . . 103
 Extended description 104
- 1996-11-28 ltvers.dtx v1.0g
 General: Check for old format
 modified /2319 33
- 1996-12-06 ltdirchk.dtx v1.0u
`\IfFileExists:` *** removed from
 various messages for GNU
 Make. internal/2338 10
- 1996-12-06 ltfloat.dtx v1.1r
`\@caption:` Call `\@setminpage` if
 needed. latex/2318 456
- 1996-12-06 ltffsini.dtx v3.0h
 General: (DPC) Remove *** from
 messages internal/2338 266
- 1996-12-17 ltclass.dtx v1.0w
`\g@addto@macro:` Use `\begingroup`
 to save making a mathord . . 570
- 1996-12-20 ltsect.dtx v1.0z
`\@dottedtocline:` Added
`\nobreak` for latex/2343 . . . 451
- 1997-01-08 fontdef.dtx v2.2q
 General: Use
`\DeclareMathDelimiter` to set
 delimiter codes 277
`\mathparagraph:` Define using
`\DeclareMathSymbol` 286
- 1997-01-08 ltfiles.dtx v1.1j
`\@include:` reset `\deadcycles`
 latex/2365 95
- 1997-01-08 ltmath.dtx v1.1d
`\root:` (DPC) Remove spurious
 space tokens from plain \TeX
 definition /2359 353
- 1997-02-05 ltclass.dtx v1.0x
`\g@addto@macro:` missing percent
 /2402 570
- 1997-02-21 ltlists.dtx v1.0r
`\@item:` `\ifvoid` check added for
`\noindent.` latex/2414 375
- 1997-03-21 ltcounts.dtx v1.1e
`\fnsymbol:` Use `\mathsection` and
`\mathparagraph.` latex/2445 149
- 1997-04-14 ltfiles.dtx v1.1k
`\document:` Set the document
 space factor defaults.
 latex/2404 91
`\normalsfcodes:` Macro added
 (from patch file) latex/2404 . . 93
- 1997-04-14 ltoutput.dtx v1.2b
`\@writsetup:` Call
`\normalsfcodes` (from patch
 file) latex/2404 510
 Move `\label` and `\index` (from
 patch file) 510
- 1997-04-24 ltbibl.dtx v1.1m
`\@citex:` `\@empty` to avoid
 primitive error on empty cite
 keys. latex/2432 475
- 1997-04-30 ltoutenc.dtx v1.9a
 General: Changed `\textsc` to
`\scshape` 116
 Introduced `\textcopyright` and
 modified `\copyright` 115
 Introduced `\textcopyright` and
 modify `\copyright` 116
 Modified `\textunderscore`,
 removing `\mathunderscore` . 116
 Modified `\underscore`,
 removing `\mathunderscore` . 116
- 1997-04-30 ltoutenc.dtx v1.9b
 General: Added `\leavevmode` to
`\textunderscore` 116
- 1997-05-04 ltoutenc.dtx v1.9c
 General: Added ‘hex index tabs’ . 122
 Added TS1 encoding v2.2.beta 128
- 1997-05-07 ltoutenc.dtx v1.9d
 General: Added `\leavevmode` to
`\textcompwordmark` 116
- 1997-05-07 ltspc.dtx v1.2v
`\newline:` Made completely
 robust. 76
- 1997-05-29 ltfsstrc.dtx v3.0j
 General: Replaced `\` by
`\MessageBreak`, as suggested
 by Donald Arseneau. 201
- 1997-05-29 ltlogos.dtx v1.1f
`\LaTeXe:` Added `\m@th` so that the
 \LaTeX 2_ϵ logo works with
 non-zero values of
`\mathsurround.` 87
- 1997-06-16 ltdirchk.dtx v1.0v
 General: documentation
 improvements mainly from

internal/2520	1	1997-10-20 ltfinal.dtx v1.1a	
1997-06-16 ltfloat.dtx v1.1s		<code>\@uclclist</code> : Removed <code>\aa</code> and <code>\AA</code>	
General: documentation fixes . . .	453	from <code>\@uclclist</code> as these are	
1997-06-16 ltfntcmd.dtx v3.3v		macros.	624
General: Fix typo in		1997-10-21 ltdefs.dtx v1.2z1	
documentation.	292	<code>\renewcommand</code> : Use	
1997-08-05 ltoutenc.dtx v1.9e		<code>\begingroup</code> / <code>\endgroup</code> rather	
General: Corrected order of		than braces for grouping, to	
arguments in <code>\UseTextSymbol</code>		avoid generating empty math	
example.	103	atom.	41
1997-08-29 ltoutenc.dtx v1.9f		1997-10-21 ltssbas.dtx v3.0t	
General: Added OT4 encoding,		<code>\define@newfont</code> : Move	
provided by Marcin Woliński. . . .	102	<code>\makeatletter</code> to	
1997-09-09 ltdefs.dtx v1.2z		<code>\nfss@catcodes</code>	164, 187
<code>\providecommand</code> : Use		<code>\nfss@catcodes</code> : Moved	
<code>\begingroup</code> to avoid		<code>\makeatletter</code> from	
generating math ords if used in		<code>\try@load@font@shape</code>	165, 188
math mode. pr/2573	43	1997-11-09 ltoutput.dtx v1.2c	
1997-09-15 ltpictur.dtx v1.1g		<code>\@specialoutput</code> : Remove	
<code>\@getcirc</code> : Warn if lines become		incorrect code: only one	
invisible pr/2524	430	<code>\@emptycol</code> is needed here . .	499
<code>\@picture@warn</code> : Macro added		<code>\@topnewpage</code> : Documentation of	
pr/2524	430	vsize check enhanced	496
<code>\@sline</code> : Warn if lines become		1997-11-13 ltssdcl.dtx v3.0f	
invisible pr/2524	421	<code>\DeclareSymbolFont</code> : (DPC)	
1997-10-06 ltcounts.dtx v1.1f		Really update <code>\group@list</code>	
<code>\@Roman</code> : Change <code>\@Roman</code> to be		dont leave new version in	
fully expandable, so that the		<code>\toks@</code> . latex/2661	234
result is written properly to		<code>\stepcounter</code> : (DPC) Remove as	
files.	149	never used. (Re)defined in	
<code>\@slowromancap</code> : Macro added. .	149	ltcounts	228
1997-10-08 ltlogos.dtx v1.1h		1997-11-19 ltfloat.dtx v1.1t	
<code>\LaTeX</code> : Simplify macro (force		<code>\@footnotetext</code> : Missing percent,	
loading of suitable math fonts		again	470
once).	87	1997-11-19 ltoutput.dtx v1.2d	
1997-10-10 ltclass.dtx v1.0y		<code>\@vtryfc</code> : Reindent code, to be	
<code>\endfilecontents</code> : <code>\@currenvir</code>		understandable(DPC).	515
in banner	574	1997-11-20 ltssdcl.dtx v3.0g	
<code>\reserved@c</code> not <code>\verbatim@out</code>		<code>\document@select@group</code> : (DPC)	
to save a csname	573	inline use of <code>\stepcounter</code>	
Check for text before or after		(faster, and saves a csname per	
<code>\end</code> environment. latex/2636 . . .	574	math version as no reset list) . .	230
Use <code>\@gobbletwo</code>	573	<code>\select@group</code> : (DPC) inline use	
1997-10-17 ltfntcmd.dtx v3.3w		of <code>\stepcounter</code> (faster, and	
<code>\check@nocorr@</code> : Check for		saves a csname per math	
vertical mode moved here, from		version as no reset list)	228
<code>\DeclareTextFontCommand</code> (see		1997-11-23 ltoutenc.dtx v1.9g	
PR/2646).	296	General: Use <code>\textperthousand</code> ,	
<code>\DeclareTextFontCommand</code> :		<code>\textpertenthousand</code> and	
Reinstalled <code>\check@icr</code> as		<code>\textfractionsolidus</code> not	
check is now done in		<code>\textpermill</code> ,	
<code>\check@nocorr@</code> (see		<code>\textpertenmill</code> and	
PR/2646).	294	<code>\textfraction</code> . /2673	128

- 1997-12-17 ltoutenc.dtx v1.9h
 General: Added `\textperthousand`
 and `\textpertenthousand` . . . 120
 Added code for textcomp.sty. . . 142
 Added section. 142
 Added textcomp.sty. 102
 As in OT1, Added `\leavevmode`
 at start of `\c`, otherwise the
 output routine might be
 invoked within the macro. . . 120
 Changed to decimal codes in
`\ooalign`. 130
 Changed to decimal codes. . . . 126
 Documentation changes and
 additions. 102
 Example corrected, braces
 removed. 102
 Removed default settings, see
 next section. 128
- 1997-12-19 ltoutenc.dtx v1.9i
 General: Documentation
 corrections. 102
- 1997-12-20 fontdef.dtx v2.2s
 General: Added documentation . 270
- 1997-12-31 ltoutenc.dtx v1.9k
 General: Further correction . . . 103
- 1998-01-12 ltoutenc.dtx v1.9k
 General: Added `\ProvidesPackage`
 for textcomp.sty 102
 Adding missing braces and
`\ushape`. 130
- 1998-01-16 ltoutenc.dtx v1.9m
 General: fixed decimal codes.
 latex/2734 126
- 1998-03-04 ltdefns.dtx v1.2z2
`\@xargdef`: Unnecessary
`\expandafter` removed:
 pr/2758 39
- 1998-03-05 ltoutenc.dtx v1.9n
 General: Added masc/fem ords as
 in pr/2579 116
- 1998-03-20 ltdefns.dtx v1.2z3
`\@thirdofthree`: Macro added . . 44
- 1998-03-20 ltoutenc.dtx v1.9o
 General: Documentation added
 about order of decls 105
 Documentation added for
 pr/2783 104
`\UndeclareTextCommand`: Macro
 added for pr/2783 113
- 1998-03-20 lttextcomp.dtx v1.9o
 General: Added various
`\UndeclareTextCommand`
 declarations for pr/2783 . . . 328
- Load decls after defaults for
 speed. 328
- 1998-03-21 ltclass.dtx v1.0z
 General: Added to documentation
 of filecontents 553
- 1998-03-21 ltclass.dtx v1.1a
`\@providesfile`: Allow &.
 Internal/2702 562
 General: Correct to new
 onlypreamble command list . 585
- 1998-03-25 ltffssbas.dtx v3.0u
`\showhyphens`: Suppress
 unnecessary error when used in
 preamble 173, 196
- 1998-04-11 fontdef.dtx v2.2t
 General: Added `\mathring` accent
 (pr2785) 284
- 1998-04-15 fontdef.dtx v2.2u
 General: Use new syntax for
`\DeclareMathDelimiter` . . . 277
- 1998-04-15 ltffssdcl.dtx v3.0h
`\@xxDeclareMathDelimiter`:
 Macro added (pr/2662) 244
- 1998-04-17 fontdef.dtx v2.2v
 General: Reinsert symbol defs for <
 and > chars. 277
- 1998-04-18 fontdef.dtx v2.2w
 General: Reinsert symbol def for /
 char. 278
- 1998-05-07 ltclass.dtx v1.1b
`\@fileswithoptions`: Modify help
 message for latex/2805 569
- 1998-05-18 lttab.dtx v1.1j
`\@endpbox`: Use `\setlength` to set
`\hsize`, so that the changes in
 the calc package apply here. 413
`\tabular*`: Use `\setlength`, so
 that calc extensions apply. . . 403
- 1998-05-20 ltfinal.dtx v1.1b
 General: Set up lccodes before
 loading hyphenation files:
 pr/2639 617
 Set up uc/lccodes after loading
 hyphenation files: pr/2639 . . 623
- 1998-05-28 lterror.dtx v1.2n
`\@notdefinable`: Added message
 re ‘end...’ pr/1555 65
- 1998-06-04 ltboxes.dtx v1.1c
`\@rule`: Support calc-expressions 389
- 1998-06-12 ltoutenc.dtx v1.9p
 General: Corrected 130 and 131,
 see pr/2834 131
 Renamed `\textmacron` pr/2840 132

1998-06-12 ltoutenc.dtx v1.9q	1999-01-05 fontdef.dtx v2.2x
<code>\add@accent</code> : Explicitly set	General: Need special protection
<code>\spacefactor</code> after <code>\accent</code>	for character <code>></code> in <code>\changes</code>
(pr/2877) 109	entry. 268
1998-06-12 lttextcomp.dtx v1.9p	1999-01-06 ltffsbas.dtx v3.0w
General: Renamed <code>\textmacron</code>	<code>\DeclareFontEncoding</code> : Added
pr/2840 324	<code>\LastDeclaredEncoding</code> to
1998-06-18 lttab.dtx v1.1k	support cyrillic integration
General: Small addition to	(pr/2988) 156, 179
documentation 392	<code>\LastDeclaredEncoding</code> : Added
1998-07-06 lttab.dtx v1.1l	<code>\LastDeclaredEncoding</code> to
General: Small correction to	support cyrillic integration
documentation 392	(pr/2988) 156, 179
1998-08-17 ltboxes.dtx v1.1e	1999-01-06 ltoutenc.dtx v1.9r
General: (RmS) Minor	<code>\@strip@args</code> : New impl for
Documentation fixes. 379	latex/2930 111
1998-08-17 ltclass.dtx v1.1c	General: Minor documentation
General: (RmS) Minor	fix. 130
documentation fixes. 553	1999-01-06 ltoutput.dtx v1.2e
1998-08-17 ltdirchk.dtx v1.0w	<code>\@makecol</code> : Added negative vskip,
General: (RmS) Documentation	as when processing outputbox
improvements. 1	below: suggested by Fred
1998-08-17 ltfntcmd.dtx v3.3x	Bartlett pr/2892 504
General: (RmS) Minor	1999-01-07 ltdefns.dtx v1.3a
documentation fixes. 292	<code>\@ifnextchar</code> : made long 50
1998-08-17 ltffsbas.dtx v3.0v	<code>\@newenvb</code> : made long and brace
General: (RmS) Documentation	optional arg. latex/2896 42
fixes. 153, 176	<code>\@testopt</code> : made long and brace
1998-08-17 ltffsdel.dtx v3.0i	optional arg. latex/2896 39
General: (RmS) Corrected minor	1999-01-07 ltdefns.dtx v1.3b
glitches in changes entries. . . 226	<code>\@ifnextchar</code> : extra <code>\long</code> .
1998-08-17 ltffsini.dtx v3.0i	latex/2902 50
General: (RmS) Minor	1999-01-07 ltoutenc.dtx v1.9r
documentation fixes. 252	General: Hackery to allow using
1998-08-17 ltlogos.dtx v1.1i	fontenc several times 144
General: (RmS) Minor	Hackery to temp support cyrillic
documentation fixes. 87	uc/lc 142
1998-08-17 ltmath.dtx v1.1c	1999-01-13 ltoutenc.dtx v1.9s
General: (RmS) Minor	<code>\@strip@args</code> : Simplified solution
documentation fixes. 351	for latex/2930 111
1998-08-17 ltmiscen.dtx v1.1g	1999-01-18 ltdefns.dtx v1.3c
General: (RmS) Minor	<code>\@yargd@f</code> : New implementation
documentation fixes. 336	DPC /2942 40
1998-08-17 ltspc.dtx v1.2w	1999-02-09 ltdefns.dtx v1.3d
General: Documentation fixes. . . 71	<code>\@yargd@f</code> : catch bad argument
1998-08-17 preload.dtx v2.1g	forms by re-inserting <code>#3</code> 40
General: (RmS) Minor	1999-02-12 lttextcomp.dtx v3.0j
documentation fixes. 288	<code>\legacyoldstylenums</code> : Use
1998-09-19 ltoutenc.dtx v1.9r	<code>\rmdefault</code> instead of <code>cmm</code>
<code>\a</code> : Added <code>\string</code> (pr/2878) . . . 114	(pr/2954) 301
1998-11-13 lttab.dtx v1.1m	1999-02-24 ltoutenc.dtx v1.9t
<code>\@array</code> : Check for hmode to see if	General: Corrected hackery cyrillic
something went wrong during	uc/lc list 142
parsing (pr/2884) 404	

1999-03-01 ltdefns.dtx v1.3e	2000-01-15 ltpictur.dtx v1.1i
\@ifnextchar: remove extra	\@upvector: Removed space at
\long. internal/2967 50	end-of-line, CAR 423
1999-04-15 ltpictur.dtx v1.1h	2000-01-30 ltfntcmd.dtx v3.3y
\@getlarrow: Replaced octal	\DeclareTextFontCommand: Use
number, CAR 422	\hmode@bgroup now (pr/3160) 294
\@upvector: Replaced octal	2000-01-30 ltoutenc.dtx v1.9y
number, CAR 423	General: Use \hmode@bgroup
General: Replaced octal number,	where applicable (pr/3160) .
CAR 422, 423 118–120, 125, 127, 128, 130
Replaced octal numbers, CAR 414	\add@accent: Use \hmode@bgroup
1999-04-19 ltfloat.dtx v1.1u	where applicable (pr/3160) . 108
\caption: Made caption an error	\hmode@bgroup: Macro added . . . 109
outside a float: latex/2815 . . 456	2000-01-30 ltoutenc.dtx v1.9z
1999-04-27 ltboxes.dtx v1.1f	\@use@text@encoding: Macro
\@parboxto: (CAR) Changed	reimplemented (pr/3160) 111, 112
\@empty to \relax as flag for	\add@accent: Macro
natural width: pr/2975 386	reimplemented (pr/3160) . . . 108
1999-04-29 ltdefns.dtx v1.3f	\hmode@start@before@group:
\@yard@f: Full expansion and	Macro added (pr/3160) 112
conversion needed for digit in	2000-05-19 ltmiscen.dtx v1.1i
new version, see pr/3013 40	\enddocument: Reset
New macro added 40	\AtEndDocument for
1999-06-10 ltoutenc.dtx v1.9u	latex/3060 337
General: Ensure that we also	2000-05-26 ltpage.dtx v1.0j
forget old options (pr/2888) . 144	\@markright: Reimplementation to
1999-06-12 ltoutenc.dtx v1.9v	fix expansion error (pr/3203). 480
General: Extend \@uclclist only	\leftmark: Use \@empty instead of
once 144	brace group (pr/3203). 480
1999-10-09 ltmath.dtx v1.1e	\markright: Reimplementation to
\active@math@prime: Macro	fix expansion error (pr/3203). 479
added, see PR 3104. 356	\rightmark: Use \@empty instead
\prime@s: Introduce	of brace group (pr/3203). . . 480
\active@math@prime. 356	2000-06-02 ltpage.dtx v1.0k
1999-10-09 ltoutput.dtx 1.2f	\@markright: Small adjustment to
\@activechar@info: Reset	give slightly less expansion,
definition of active prime	CAR 480
character (used in math	\markright: Small adjustment to
mode) 508	give slightly less expansion,
1999-10-28 ltoutenc.dtx v1.9w	CAR 479
\add@accent: Give	Tidied 1.0j reimplementation,
\accent@spacefactor a	CAR 479
default definition (pr/3084) . 109	2000-07-11 ltmiscen.dtx v1.1j
1999-12-08 ltoutenc.dtx v1.9x	\enddocument: Fix typo in
General: Changed \CYRRHOOK and	warning 338
\cyrrhook to \CYRRHK and	2000-07-12 ltoutput.dtx 1.2g
\cyrrhk as name changed in	General: Ensure that rule is in
the cyrillic bundle for naming	\normalcolor 547
consistency with other “hook”	2000-07-12 ltoutput.dtx 1.2i
glyphs. 142	\@makecol: Removed negative
2000-01-07 ltmiscen.dtx v1.1h	vskip, as it gives unacceptable
\@verbatim: Disable hyphenation	results when the depth is large:
even if the font allows it. . . . 347	pr/3189 504

2000-07-19 ltoutput.dtx v1.2h	Guard against math active equal sign in <code>\Relbar</code> (pr/3333) . . .	282
<code>\@writesetup</code> : Reset and restore <code>\@if@newlist</code> for internal/3231	2001-06-04 ltclass.dtx v1.1e	
2000-08-23 ltfinal.dtx v1.1c	<code>\@providesfile</code> : But only if it is a char (pr/3334)	562
General: Fix typo in warning . . .	2001-06-04 ltdirchk.dtx v1.0y	
2000-08-30 ltoutenc.dtx v1.91	General: But only if it is a char (pr/3334)	4
<code>\@use@text@encoding</code> : Rearranged but no change to final code, CAR (pr/3160)	2001-06-04 ltpictur.dtx v1.1j	
<code>\add@accent</code> : Rearranged but no change to final code, CAR (pr/3160)	<code>\@sline</code> : Don't warn for exactly zero pr/3318	421
2000-09-01 ltfinal.dtx v1.1d	2001-06-04 ltvers.dtx v1.0i	
<code>\errhelp</code> : Set error help empty at very end (pr/449 done correctly).	General: Check for old format disabled	33
2000-09-24 ltfloat.dtx v1.2b	2001-06-05 ltoutenc.dtx v1.94	
<code>\end@dblfloat</code> : FMI: use output routine to defer float	General: Text composite Commands need kludges for ‘,’ – see tlb1903.lvt	119
2000-09-24 ltoutput.dtx v1.2b	2001-08-26 ltclass.dtx v1.1f	
<code>\docclearpage</code> : FMI: ensure <code>\docclearpage</code> is called again until all floats are output. . .	<code>\@providesfile</code> : Readded setting of space char (pr/3353)	562
2000-09-24 ltoutput.dtx v1.2n	2002-02-24 ltplain.dtx v1.1x	
<code>\@addtocurcol</code> : FMI: test for wide float was in wrong place . . .	<code>\loggingall</code> : Macro added	30
2001-01-07 ltoutput.dtx v1.2j	<code>\loggingoutput</code> : Macro added . .	30
<code>\@writesetup</code> : And do it in the right macro (pr/3286)	<code>\showoutput</code> : Use newly added <code>\loggingoutput</code>	30
2001-02-16 ltxref.dtx v1.1k	<code>\tracingall</code> : Use newly added <code>\loggingoutput</code>	30
<code>\@newl@bel</code> : Added an extra grouplevel (PR3250), jlb . . .	2002-06-16 ltoutenc.dtx v1.95	
2001-05-25 ltclass.dtx v1.1d	General: Added <code>\textbardbl</code> (pr/3400)	125
<code>\@providesfile</code> : Explicitly set catcode of <code>\endlinechar</code> to 10 (pr/3334)	Added default for <code>\textbardbl</code> (pr/3400)	115
2001-05-25 ltdirchk.dtx v1.0x	2002-06-17 ltoutenc.dtx v1.95	
General: Explicitly set catcode of <code>\endlinechar</code> to 10 (pr/3334) . .	General: Corrected <code>\c</code> for T1 (pr/3442)	120
2001-05-28 ltoutenc.dtx v1.93	Definition of <code>\textexclamdown</code> changed (pr/3368)	118
General: Added composites for compatibility with T1, pr/3295	Definition of <code>\textquestiondown</code> changed (pr/3368)	118
Changed the effect of <code>\.i</code> , pr/3295	2002-06-18 ltoutenc.dtx v1.95	
2001-06-02 fontdef.dtx v2.2y	General: Changed def for <code>\textregistered</code> to avoid small caps (pr/3420)	116
General: Provide default cfg files (pr/3264)	2002-10-01 ltfloat.dtx v1.1v	
2001-06-04 fontdef.dtx v2.2z	<code>\thempfootnote</code> : Use braces around <code>\itshape</code> to keep font change local (pr/3460).	469
General: Guard against math active equal and pipe sign in <code>\models</code> (pr/3333)	2002-10-02 ltfsbas.dtx v3.0x	
	<code>\DeclareFontSubstitution</code> : Adding <code>\LastDeclaredEncoding</code> introduced a bug as on some	

occasions that macro name was stored in the internal lists instead of the actual encoding. (pr/3459)	156, 179	<code>\@xargdef</code> : Use kernel version of <code>\@ifnextchar</code> (pr/3501)	39
2002-10-28 ltlists.dtx v1.0s		<code>\@xdblarg</code> : Use kernel version of <code>\@ifnextchar</code> (pr/3501)	51
<code>\endtrivlist</code> : Check for math mode (pr/3437)	373	2004-01-23 ltdefns.dtx v1.3g	
2002-10-28 ltoutenc.dtx v1.96		<code>\kernel@ifnextchar</code> : Added macro (pr/3501)	50
General: coding change, to follow bug fix by DEK in plain.tex (pr/3469)	119, 127	2004-01-28 ltclass.dtx v1.1g	
2002-12-13 ltbibl.dtx v1.1n		<code>\@providesfile</code> : Use kernel version of <code>\@ifnextchar</code> (pr/3501)	562
<code>\@citex</code> : Added <code>\leavevmode</code> in case citation is at start of paragraph (pr/3486)	475	2004-01-28 ltvers.dtx v1.0k	
2003-01-01 ltfntcmd.dtx v3.3z		General: Check for old format made 5 years (pr/3601)	33
General: Code checked and documentation extended by Chris	294	2004-02-02 fontdef.dtx v2.3	
2003-05-18 ltbibl.dtx v1.1o		General: Many things from here on made robust	282
<code>\nocite</code> : Check if we are after <code>\document</code>	476	2004-02-02 ltoutenc.dtx v1.99	
2003-08-27 ltpictur.dtx v1.1k		General: Added <code>\textbigcircle</code>	125
<code>\@bezier</code> : added missing displacement pr/3566	435	2004-02-04 fontdef.dtx v2.3a	
<code>\@sline</code> : check for <code>\@linechar</code> being empty pr/3570	421	General: Added bigtriangle synonyms for stmaryrd	280
2003-10-13 ltfinal.dtx v1.1e		2004-02-04 ltspace.dtx v1.3	
General: Added extra <code>\lccode</code> for <code>\-</code> and <code>\textcompwordmark</code>	618	<code>\nbreakdashes</code> : (Macro added	84
2003-12-16 ltoutput.dtx v1.2k		2004-02-06 ltoutenc.dtx v1.99d	
<code>\@makecol</code> : Ensure that <code>\@elt</code> has a defined state (pr/3586)	505	<code>\@inmathwarn</code> : New command added to fix severe bug: pr/3563	106
2003-12-30 ltpictur.dtx v1.1j		2004-02-07 ltoutput.dtx v1.2l	
<code>\@getcirc</code> : issue warning if circle size can't be met pr/3473	430	<code>\@doclearpage</code> : Empty kludgeins box if necessary, pr/3528	503
2004-01-03 ltoutenc.dtx v1.99b		2004-02-13 ltoutenc.dtx v1.99e	
General: Added <code>\textogonekcentered</code> (pr/3532)	120	General: Documentation fixes: typos	102
Added composites for <code>\k</code> (pr/3532)	125	2004-02-15 ltbibl.dtx v1.1q	
Use <code>\ooalign</code> for <code>\k</code> (pr/3532)	120	<code>\@cite@ofmt</code> : Added hook with default value <code>\hbox</code>	477
2004-01-04 ltbibl.dtx v1.1p		<code>\@citex</code> : Changed to use a hook with default value <code>\hbox</code>	476
<code>\nocite</code> : Changed error message	476	2004-02-15 ltspace.dtx v1.3a	
2004-01-04 ltoutenc.dtx v1.99c		<code>\nbreakdashes</code> : (Added spacefactor setting	84
General: More adjustments for <code>ogonek</code> (pr/3532)	120	2004-10-20 ltoutput.dtx v1.2m	
2004-01-23 ltdefns.dtx v1.1g		<code>\@makecol</code> : Removed dead code	504
<code>\@newenva</code> : Use kernel version of <code>\@ifnextchar</code> (pr/3501)	42	2005-07-27 ltfssdcl.dtx v3.0j	
<code>\@testopt</code> : Use kernel version of <code>\@ifnextchar</code> (pr/3501)	40	<code>\DeclareMathAlphabet</code> : (MH) Make document commands robust	236
		<code>\DeclareSymbolFontAlphabet</code> : (MH) Make document commands robust	249
		<code>\new@mathalphabet</code> : (MH) Make document commands robust	237

<code>\non@alpherr:</code> (MH) Change because command is now properly robust	230	<code>\t@st@ic:</code> Use switch <code>\ifmaybe@ic</code> instead of <code>\if@tempswa</code>	297
<code>\SetMathAlphabet:</code> (MH) Make document commands robust	238	2010-08-17 ltmiscen.dtx v1.1k	
2005-09-27 ltoutenc.dtx v1.99g		<code>\enddocument:</code> Use braces around <code>\input</code> arg (pr/4124)	338
General: Replace <code>\sh@ft</code> by <code>\ltx@sh@ft</code>	118, 120, 127	2010-08-17 ltmiscen.dtx v1.1l	
2005-09-27 ltplain.dtx v1.1y		<code>\enddocument:</code> Change of plan: use <code>\@input</code> instead (pr/4124)	338
<code>\ltx@sh@ft:</code> New macro	29	2011-05-08 ltfssdcl.dtx v3.0n	
<code>\sh@ft:</code> Macro no longer used but left for compatibility	29	<code>\in@:</code> Simplified thanks to Bruno.	226
2005-11-08 ltoutenc.dtx v1.99h		2011-08-19 ltclass.dtx v1.1i	
General: Added <code>\ij</code> and <code>\IJ</code> from babel. (pr/3771)	115, 119, 121	<code>\@ifclasswith:</code> Re-jig definition after more stringent <code>\in@</code> test.	560
2005-11-10 ltmath.dtx v1.1g		2011-09-03 ltfssdcl.dtx v3.0o	
<code>\l:</code> (MH) Fixed potential problem in <code>\l</code> (pr/3399).	357	<code>\new@mathversion:</code> (Will) Remove <code>\global</code> before <code>\newcount</code> (unnecessary and caused etex bug).	233
General: (MH) Minor documentation fixes.	351	2012-01-20 ltplain.dtx v2.0b	
2006-05-18 ltboxes.dtx v1.1g		<code>\loggingall:</code> etex tracing if available	30
<code>\@parboxto:</code> Ensure <code>\@parboxto</code> holds the value of <code>\@tempdimb</code> not the register itself (pr/3867)	386	2013-07-07 ltclass.dtx v1.1i	
2006-09-13 ltoutput.dtx v1.1m		General: Correctly describe how the date in <code>\@ifpackagelater</code> is used	556
General: Ensure that rule is in <code>\normalcolor</code>	548	2014-04-18 ltoutput.dtx v1.1o	
2007-08-05 ltclass.dtx v1.1h		General: Handle infinite glue from <code>\enlargethispage</code> (pr/4023)	548
<code>\@fileswithoptions:</code> Prevent loss of brackets PR/3965	567	2014-04-24 ltoutput.dtx v1.2n	
2007-08-06 ltcntrl.dtx v1.0h		<code>\fl@tracemessage:</code> Renamed internal trace commands; provide as package	538
<code>\@fornoop:</code> Really make defs long	58	2014-04-27 ltfloat.dtx v1.2b	
2007-08-31 ltfssdcl.dtx v3.0l		<code>\end@dblfloat:</code> Inline the code to allow some coexistence with packages that hook into <code>\end@float</code> and do not know about the algorithm change	461
<code>\SetSymbolFont@:</code> Font warning changed to info for encoding change (pr/3975)	235	2014-06-10 ltfloat.dtx v1.2b	
2009-09-24 ltvers.dtx v1.0l		<code>\end@dblfloat:</code> missing <code>\fi</code> added	461
General: Stop checking for old format	33	2014-12-30 ltfinal.dtx v2.0a	
2009-10-20 ltfssdcl.dtx v3.0m		<code>\newmarks:</code> macro added	613
<code>\in@:</code> More robust thanks to Heiko.	226	<code>\newXeTeXintercharclass:</code> macro added	613
2009-10-28 lttextcomp.dtx v1.99k		2014-12-30 ltfloat.dtx v1.2a	
General: Added Latin Modern and TeX Gyre subsets	330	<code>\@textsubscript:</code> Command added (latexrelease)	470
2009-11-04 lttextcomp.dtx v1.99l		<code>\textsubscript:</code> Command added (latexrelease)	470
General: Added more Latin Modern and TeX Gyre subsets	330	2014-12-30 ltfssbas.dtx v3.0y	
2009-12-14 ltfntcmd.dtx v3.4a		<code>\mathgroup:</code> move allocation to ltplain.	153, 176
<code>\ifmaybe@ic:</code> Macro added	297		
<code>\maybe@ic:</code> Use switch <code>\ifmaybe@ic</code> instead of <code>\if@tempswa</code>	297		

2014-12-30 ltoutput.dtx v1.2m	2015-01-09 ltspacedtx v1.1h
General: Command updated	<code>\addpenalty</code> : Donald Arseneau's
(latexrelease) 547	fix from PR/377703
2014-12-30 ltplain.dtx v2.0a	(latexrelease) 81
<code>\e@alloc</code> : macro added 20	2015-01-10 ltcountsdtx v1.1h
<code>\e@alloc@chardef</code> : macro added 19	<code>\@fnsymbol</code> : Unse <code>\TextOrMath</code>
<code>\e@alloc@top</code> : macro added 19	(latexrelease) 150
<code>\e@ch@ck</code> : macro added 20	<code>\@stpelit</code> : Reset all within
<code>\extrafloats</code> : macro added 20	counters in one go
<code>\newlanguage</code> : New engine-specific	(latexrelease) 147
allocation scheme	2015-01-11 ltcountsdtx v1.1h
(latexrelease) 18	<code>\TextOrMath</code> : Add command to
2014-12-30 ltspacedtx v1.3b	solve robustness issues
<code>\@</code> : <code>\@</code> discards spaces when	(pr/3752) (latexrelease) 150
moving (pr3039)(latexrelease) 85	2015-01-11 ltfloat.dtx v1.2b
2015-01-03 ltdefns.dtx v1.4a	<code>\dbfloatplacement</code> : float order
<code>\typein</code> : use modified definition in	in 2-column (latexrelease) 462
luatex 38	<code>\xfloat</code> : Check for valid option
2015-01-03 ltdirchk.dtx v1.1	(latexrelease) 457
General: Enable extra primitives	<code>\enddbfloat</code> : float order in
when LuaTeX is used 3	2-column (latexrelease) 461
2015-01-03 ltfinal.dtx v2.0a	2015-01-11 ltssbas.dtx v3.0y
General: Skip resetting codes with	<code>\DeclareMathSizes</code> : Allow
Unicode engines 623	arbitrary units
Unicode data loading added 615	(latexrelease) 158, 181
2015-01-07 ltvers.dtx v1.0n	2015-01-11 ltspacedtx v1.3d
<code>\IncludeInRelease</code> : macro added 34	<code>\Esphack</code> : Allow hyphenation
2015-01-08 ltboxes.dtx v1.1h	(Donald Arseneau pr/3498)
<code>\framebox</code> : Make Robust	(latexrelease) 79
(latexrelease) 384	<code>\esphack</code> : Allow hyphenation
<code>\makebox</code> : Make Robust	(Donald Arseneau pr/3498)
(latexrelease) 380	(latexrelease) 78
<code>\parbox</code> : Make Robust	2015-01-14 ltoutput.dtx v1.2n
(latexrelease) 385	<code>\@addtocurcol</code> : float order in
<code>\raisebox</code> : Make Robust	2-column (latexrelease) 519
(latexrelease) 390	<code>\@addtodblcol</code> : float order in
<code>\rule</code> : Make Robust	2-column (latexrelease) 530
(latexrelease) 389	<code>\@addtonextcol</code> : float order in
<code>\savebox</code> : Make Robust	2-column (latexrelease) 526
(latexrelease) 382	<code>\doclearpage</code> : Empty kludgeins
2015-01-08 ltdefns.dtx v1.4a	box if necessary, pr/3528 502
<code>\MakeRobust</code> : Added macro 47	float order in 2-column
2015-01-08 ltlength.dtx v1.1c	(latexrelease) 502
<code>\setlength</code> : to ensure first length	<code>\startdblcolumn</code> : float order in
argument is terminated.	2-column (latexrelease) 514
(latexrelease) 152	<code>\xtryfc</code> : float order in 2-column
2015-01-08 ltmath.dtx v1.1h	(latexrelease) 516
<code>\]</code> : Make Robust (latexrelease) 357	<code>\ztryfc</code> : float order in 2-column
<code>\]</code> : Make Robust (latexrelease) 357	(latexrelease) 517
2015-01-09 ltssini.dtx v3.1a	2015-01-14 ltspacedtx v1.3e
<code>\em</code> : Allow <code>\emph</code> to produce small	<code>\addpenalty</code> : Avoid adding
caps (latexrelease) 263	redundant skips (DPC) 82
<code>\minnershape</code> : macro added	
(latexrelease) 263	

2015-01-17 ltvers.dtx v1.0m	2015-02-21 ltfsscmp.dtx v3.0d
\IncludeInRelease: modified with	General: Removed autoload code 222
\currname 34	2015-02-21 ltfssdcl.dtx v3.0p
2015-01-19 ltvers.dtx v1.0o	General: Removed autoload code 226
\IncludeInRelease: Optional	2015-02-21 ltfsstrc.dtx v3.0k
argument 34	General: Removed autoload code 199
2015-01-20 ltoutput.dtx v1.2m	2015-02-21 ltoutenc.dtx v1.99m
\fl@tracemessage: Reset	General: Removed autoload code 102
\IncludeInRelease flags . . . 539	2015-02-21 ltoutput.dtx v1.2n
2015-01-22 ltvers.dtx v1.0p	General: Removed autoload code 481
General: Preserve any \everyjob	\fd@depth: macro
material inserted by a loader	added(latexrelease) 501
(.ini file) 34	2015-02-21 ltpictur.dtx v1.1k
2015-01-23 ltfinal.dtx v2.0b	General: Removed autoload code 414
\newmarks: use reserved count 256 613	2015-02-21 ltplain.dtx v2.0e
\newXeTeXintercharclass: use	General: Removed autoload code 15
reserved count 257 613	2015-02-21 lttab.dtx v1.1n
2015-01-23 ltplain.dtx v2.0c	General: Removed autoload code 392
\extrafloats: reserve counts	2015-02-21 ltvers.dtx v1.0r
256–265 20	General: Removed autoload code 33
2015-01-24 ltfinal.dtx v2.0c	2015-02-21 ltvers.dtx v1.0w
General: Skip T1-code entirely	\IncludeInRelease: set
with Unicode engines 615	\currname empty here (in
2015-02-03 ltfinal.dtx v2.0d	case \IncludeInRelease input
General: Set \lccode for - with	early) 34
Unicode engines 616	2015-02-22 ltfsscmp.dtx v3.0e
2015-02-16 ltoutenc.dtx v1.99m	General: Moved all code into
General: Added \textcommabelow	latexrelease - obsolete
latex/4414 117	commands are no longer
2015-02-16 ltoutenc.dtx v1.99n	automatically part of the
General: Added \textcommaabove 117	kernel 222
Added composites for ç 125	2015-03-02 ltplain.dtx v2.0f
Added composites for \c 120	\@mathgroup@top: macro added 19
2015-02-16 lttextcomp.dtx v1.99m	\newlanguage: allow 255 math
General: Added lmtt (Heiko	groups in Unicode engines . . . 18
Oberdiek) latex/4415 330	2015-03-10 ltplain.dtx v2.0g
2015-02-19 ltvers.dtx v1.0q	\hideoutput: macro added 31
\IncludeInRelease: Swap	\loggingall: Reorganise to be less
argument order 34	noisy 30
2015-02-20 ltplain.dtx v2.0d	\tracingnone: macro added 31
\loggingall: Spell commands	2015-03-12 ltoutput.dtx v1.2m
correctly :-)) 30	General: initialise \@dbldeferlist
2015-02-21 ltdefns.dtx v1.4b	again 493
General: Removed autoload	2015-03-18 ltfssdcl.dtx v3.0q
support 36	\DeclareSymbolFont: Restrict
2015-02-21 lterror.dtx v1.2o	Symbol fonts to 0-15 234
General: Removed autoload	\document@select@group:
support 60	Introduce \@mathgroup@top 230
2015-02-21 ltfiles.dtx v1.1m	\select@group: Introduce
General: Removed autoload	\@mathgroup@top 228
support 88	2015-03-26 ltfinal.dtx v2.0d
2015-02-21 ltfssbas.dtx v3.0z	General: Use renamed
General: Removed autoload	unicode-letters.def 615
code 153, 176	

2015-04-07 ltffsbas.dtx v3.1a	callback_descriptions: Function added	611
\wrong@fontshape: Try loading fd file if family has changed	\catcodetable@atletter: Macro added	595
2015-04-28 ltfinal.dtx v2.0f	\catcodetable@initex: Macro added	595
\newXeTeXintercharclass: define \xe@alloc@intercharclass for compatibility with older xelatex initialisation	\catcodetable@latex: Macro added	595
2015-05-10 ltlists.dtx v1.0t	\catcodetable@string: Macro added	595
\@doendpe: Explicitly reset \clubpenalty before clearing \everypar; see also pr/0462 and pr/4065	add_to_callback: Function added	609
2015-06-19 ltfinal.dtx v2.0g	remove_from_callback: Function added	610
\@alloc@intercharclass@top: Use -1 for first range to get contiguous allocation	new_attribute: Function added	602
\newmarks: Use -1 for first range to get contiguous allocation	disable_callback: Function added	611
2015-06-19 ltplain.dtx v2.0h	in_callback: Function added	611
General: delete spurious old definition of \newtoks	\newattribute: Macro added	594
\@alloc: extra braces in case arguments not single token	\newcatcodetable: Macro added	594
\newlanguage: Use -1 for first range to get contiguous allocation	\newluabytocode: Macro added	597
2015-06-23 ltfinal.dtx v2.0h	\newluachunkname: Macro added	597
General: set \patch@level in ltvers rather than in ltfinal/ltpatch	\newluafunction: Macro added	596
2015-06-23 ltvers.dtx v1.0t	\newwhatsit: Macro added	597
General: set \patch@level in ltvers rather than in ltfinal/ltpatch	module_error: Function added	600
2015-08-06 ltplain.dtx v2.0i	module_info: Function added	600
\extrafloats: Add \string in case argument is not an unexpandable primitive	module_warning: Function added	600
2015-08-23 ltdirchk.dtx v1.2	modules: Function modified	599
General: Do not use luatex prefix	create_callback: Function added	608
2015-08-23 ltvers.dtx v1.0v	provides_module: Function added	599
General: Allow negative patchlevel for pre-release	luatexbase: Table added	599
2015-08-30 ltplain.dtx v2.1a	2015-10-02 ltdirchk.dtx v1.2a	
\newinsert: new \newinsert implementation	General: Allow backing out of unprefixed names	3
2015-09-205 ltoutput.dtx v1.3a	2015-10-02 ltluatex.dtx v1.0b	
General: extended \@freelist	General: Fix backing out of T _E X code	598
2015-09-24 ltluatex.dtx v1.0a	2015-10-02 ltluatex.dtx v1.0c	
call_callback: Function added	General: Allow backing out of Lua code	598
callback.register: Function modified	2015-10-02 ltluatex.dtx v1.0e	
	uninstall: Function added	612
	2015-10-03 ltluatex.dtx v1.0f	
	provides_module: use luatexbase_log	599
	2015-10-27 ltplain.dtx v2.1b	
	\extrafloats: Use global assignment when switching to extended range	20

2015-11-07 ltspace.dtx v1.3f	2015-12-10 ltfinal.dtx v2.0i
<code>\@esphack</code> : Only space if there is	General: Use new common
no space at the end of the hlist	Unicode data loaders 615
latex/4443 78	2015-12-18 ltluatex.dtx v1.0l
2015-11-14 ltluatex.dtx v1.0g	General: Load Unicode data from
General: Track Lua _T _E X changes	source 595
for <code>(new)token.create</code> 601	2016-01-04 ltfinal.dtx v2.0j
2015-11-18 ltplain.dtx v2.2a	General: Do not set up inter
<code>\newlanguage</code> : Extended stream	character classes for XeTeX . . 615
allocation in luatex (0.85) . . . 18	<code>\e@alloc@intercharclass@top</code> :
2015-11-19 ltplain.dtx v2.2b	Start allocation at one not
<code>\newlanguage</code> : Only extend	three 613
allocation of write streams (see	2016-01-05 ltfinal.dtx v2.0k
luatex list) 18	<code>\e@alloc@intercharclass@top</code> :
2015-11-27 ltluatex.dtx v1.0h	Remove duplicated code . . . 613
<code>callback_descriptions</code> : Match	2016-01-05 ltfinal.dtx v2.0l
test in in-callback latex/4445 . 611	General: Correct <code>latexrelease</code>
<code>in_callback</code> : Guard against	guards 615
undefined list latex/4445 . . . 611	Ensure old definitions for
2015-11-29 ltluatex.dtx v1.0i	inter-character class toks are
General: Declare this as local	available using <code>latexrelease</code> . . 615
before used in the module error	Missing brace 615
definitions (PHG) 599	2016-01-05 ltfinal.dtx v2.0m
<code>call_callback</code> : Check name is not	General: Undefine XeTeX classes
nil in error message (PHG) . . 609	when using patching an older
<code>create_callback</code> : Check name is	kernel 615
not nil in error message	2016-01-05 ltfinal.dtx v2.0p
(PHG) 608	General: Only apply XeTeX
2015-12-02 ltluatex.dtx v1.0j	change if XeTeX is in use . . 615
General: Adjust hashtokens to	2016-02-11 ltluatex.dtx v1.0m
store the result of	General: pdf_stream_filter_callback
tex.hashtokens()), not the	removed 606
function (PHG) 601	process_rule, [hv]pack_quality
Assorted typos fixed (PHG) . . 592	append_to_vlist_filter added . . 605
Declaration/use of first_head	read_cidmap_file added 605
fixed (PHG) 600	show_warning_message added . . 606
Remove nonlocal iteration	token_filter removed 605
variables (PHG) 592	2016-02-18 ltfsdcl.dtx v3.0r
Remove unreachable code after	<code>\@DeclareMathDelimiter</code> : Check
calls to error() (PHG) 592	for delimiter not <code>\delimiter</code> . 244
2015-12-02 ltluatex.dtx v1.0k	<code>\DeclareMathAccent</code> : Check for
General: resolve name and	mathaccent not <code>\mathacemt</code> . 240
i.description (PHG) 607	<code>\DeclareMathRadical</code> : Check for
<code>call_callback</code> : Give more specific	radical not <code>\radical</code> 248
error messages (PHG) 609	<code>\DeclareMathSymbol</code> : Check for
<code>add_to_callback</code> : Give more	mathchar not <code>\mathchar</code> . . . 242
specific error messages (PHG) . 609	2016-03-13 ltluatex.dtx v1.0n
<code>remove_from_callback</code> : adjust	General: contribute_filter added . 605
initialisation of cb local	insert_local_par added 605
(PHG) 610	2016-03-29 ltpictur.dtx v1.1l
Give more specific error	<code>\@oval</code> : add setting of line tests . 431
messages (PHG) 610	initialise tests 431
<code>create_callback</code> : Give more	<code>\@ovhorz</code> : use glue not leaders if
specific error messages (PHG) . 608	horizontal line not required . . 431

<code>\@ovvert</code> : use glue not leaders if vertical line not required ...	431	2016-12-03 fontdef.dtx v3.0a	
<code>\if@ovhline</code> : macro added (latex/4452)	430	General: (DPC) Default to TU encoding for Unicode TeX engines	270
<code>\if@ovvline</code> : macro added (latex/4452)	430	<code>\shapedefault</code> : (DPC) Default to TU encoding for Unicode TeX engines	273
2016-04-22 ltfinal.dtx v2.0q		2016-12-04 ltoutenc.dtx v2.0a	
<code>\e@alloc@intercharclass@top</code> : XeTeX 0.99996 has 4096 char classes not 256	613	General: Added TU encoding ...	132
2016-06-19 ltoutenc.dtx v1.99m		2017-01-01 ltoutput.dtx v1.3b	
General: OT1 definition (was duplicate T1 definition)	120	General: make fpmin negative so ignored even if float height is negative	547
2016-06-20 ltclass.dtx v1.1j		2017-01-10 ltffsbas.dtx v3.2a	
<code>\ifclasslater</code> : don't declare as <code>\@onlypreamble</code>	559	<code>\showhyphens</code> : Add version of <code>\showhyphens</code> that works with XeTeX.	173, 196
2016-07-29 ltplain.dtx v2.2c		2017-01-23 ltoutenc.dtx v2.0b	
<code>\extrafloats</code> : use <code>\global</code> <code>\chardef</code>	21	General: Added TU specific commands in ASCII range pr/4500	132
<code>\newinsert</code> : fix for tlb-newinsert-001	22	2017-01-24 ltoutenc.dtx v2.0c	
2016-10-02 ltclass.dtx v1.2a		General: Declare TU composites for i and j	132
<code>\ifclasswith</code> : Ignore spaces while checking for option clash	560	Make <code>\textasteriskcentered</code> U+2217 not U+204E	132
<code>\ExecuteOptions</code> : Ignore spaces in argument	564	TeX ligature syntax for xetex and luatex reversed	132
2016-10-15 ltdirchk.dtx v1.2b		2017-01-24 ltoutenc.dtx v2.0d	
General: Require eTeX	4	General: Declare macron composites for YyGg	132
2016-10-15 lterror.dtx v1.2p		2017-02-12 ltoutenc.dtx v2.0e	
General: Require eTeX	60	General: Declare fallback code for <code>\textasteriskcentered</code> ...	132
2016-10-15 ltfinal.dtx v2.0r		2017-02-18 ltluatex.dtx v1.1c	
General: Require eTeX	613	<code>new_attribute</code> : Parameterise count used in tracking	602
2016-10-15 ltfinal.dtx v2.0s		<code>new_bytecode</code> : Parameterise count used in tracking	603
General: Tidy up status of char 127	613	<code>new_chunkname</code> : Parameterise count used in tracking	603
2016-10-15 ltffsini.dtx v3.1b		<code>new_whatst</code> : Parameterise count used in tracking	602
General: Require eTeX	252	2017-02-19 ltoutenc.dtx v2.0f	
2016-10-15 ltplain.dtx v2.2d		General: add <code>\@empty</code> to guard against 3rd argument being empty	119
General: Require eTeX	15	declare composites with empty base for hat and tilde, use same slots for <code>\textasciicircum</code> and <code>\textasciitilde</code>	132
2016-10-16 ltplain.dtx v2.3a		declare straight quotes using new <code>\remove@tlig</code> command	132
<code>\newlanguage</code> : Allow languages up to 16383 in luatex	18		
2016-10-19 ltcounts.dtx v1.1j			
<code>\TextOrMath</code> : Test directly for <code>\protected</code>	150		
2016-11-06 ltplain.dtx v2.3b			
General: Drop <code>\outer</code> entirely ..	15		
2016-11-09 ltclass.dtx v2.1b			
<code>\@fileswithoptions</code> : Improve <code>\ifx</code> tests PR/4497	567		
2016-11-17 ltluatex.dtx v1.0p			
General: call_edit added	606		

2017-02-22 ltoutenc.dtx v2.0g	2018-01-06 ltdefs.dtx 1.5c
General: Fix typo introduced at	\@ifundefined: Avoid defining
2.0f 132	undefined commands to \relax 49
2017-02-24 ltoutenc.dtx v2.0h	2018-02-18 ltclass.dtx v1.2d
General: introduce	\@ifclasslater: Added 0 up front
\DeclareUnicodeAccent . . . 132	to make bad data come out as
\DeclareTextCompositeCommand:	0. 559
add check whether the accent	General: Introduce rollback
command is defined for this	concept 577
encoding 109	2018-03-08 ltcounts.dtx v1.1k
2017-03-08 ltclass.dtx v1.2c	\@ifbothcounters: Interface
\@ifclasslater: add	added 148
\@parse@version@dash to	\@removefromreset: Interface
support yyyy-mm-dd as well as	added 147
yyyy/mm/dd 559	\counterwithin: Interface added 148
2017-03-09 ltfinal.dtx v2.0t	\counterwithout: Interface
\l@nohyphenation: ensure	added 148
\l@nohyphenation is defined. 618	2018-03-24 ltclass.dtx v1.2e
2017-03-09 ltmiscen.dtx v1.1m	\pkgcls@use@this@release: Use
\@verbatim: Use \language not	full file name for old release . 582
\hyphenchar 347	2018-03-25 ltfinal.dtx v2.1a
\verb: Use \language to stop	General: default to UTF-8 619
hyphenation 349	\UseRawInputEncoding: Macro
2017-03-10 ltfiles.dtx v1.1n	added 620
\document: Save language default 91	2018-03-27 ltclass.dtx v1.2f
2017-03-10 ltoutput.dtx v1.3c	\endfilecontents: Use full file
\@writsetup: Reset \language . 508	name for old release 574
2017-03-13 ltdefs.dtx v1.5a	2018-04-06 ltfinal.dtx v2.1b
\ -: Define \- in terms of	\UseRawInputEncoding: Undo
\hyphenchar 51	changes to
2017-03-27 ltdefs.dtx v1.5b	\DeclareFontEncoding@ and
\@dischyp: Define \@dischyp	definition of
after \- 51	\DeclareUnicodeCharacter . 620
2017-03-28 ltluatex.dtx v1.1e	2018-04-07 ltfinal.dtx v2.1c
General: glyph_stream_provider	\UseRawInputEncoding: Undefine
added 606	\inputencodingname 620
2017-03-29 ltboxes.dtx v1.3a	2018-04-08 ltclass.dtx v1.2g
\@arrayparboxrestore: Reset	\@ifclasslater: Strip leading
\lineskiplimit 387	spaces from dates. 559
2017-04-05 ltoutenc.dtx v2.0i	2018-04-08 ltclass.dtx v1.2h
\DeclareTextCompositeCommand:	\@onefilewithoptions: Pass
Declare accent command if not	expanded date 578
already declared when	2018-04-08 ltfinal.dtx v2.1d
declaring a composite. 109	General: Delay full UTF-8
2017-04-10 ltplain.dtx v2.3c	handling to \everyjob 620
\newlanguage: Correction to code	2018-04-11 ltcounts.dtx v1.1l
to skip write18 in luatex 18	\counterwithin: Correct default
2017-04-11 ltoutput.dtx v2.4a	(issue/38) 148
\newpage: account for the depth of	2018-05-02 ltluatex.dtx v1.1g
the last row of the page 495	General: find_sfd_file removed . . 605
2017-12-17 ltoutput.dtx v1.4b	finish_synctex_callback added . 606
\@addtonextcol: fix doc guards . 526	glyph_not_found added 606
	read_sfd_file removed 605

2018-05-08 ltclass.dtx v1.2i		<code>\pkgcls@parse@date@arg</code> : Make suspicious rollback a warning not error: github issue 43 . . . 580	<code>\protected@file@percent</code> : Sometimes mask the newline char when writing to files (github/73) 339
2018-05-11 ltfinal.dtx v2.18		General: Make invalid UTF-8 also safe, for legacy filesystem encodings 620	2018-09-26 ltsect.dtx v1.1c
2018-05-29 ltclass.dtx v1.2j		<code>\endfilecontents</code> : use <code>\csname</code> not <code>\@undefined</code> 575	<code>\addcontentsline</code> : Sometimes mask the newline char when writing to files (github/73) . . 450
2018-08-11 ltoutenc.dtx v2.0j		General: Provide <code>\guillemetleft</code> and <code>\guillemetright</code> . . 121, 127, 135	2018-10-10 ltspace.dtx v1.3i
2018-08-18 ltluatex.dtx v1.1h		General: append <code>_to_vlist_filter</code> is <code>exclusive</code> 605	<code>\@esphack</code> : Don't introduce breakpoints if <code>\@nobreak</code> is true and after sections 78
2018-08-24 ltfinal.dtx v2.1f		<code>\document@default@language</code> : Add to latexrelease (github/68) 618	2018-10-11 ltmiscen.dtx v1.1o
2018-09-02 ltsect.dtx v1.1b		<code>\@dottedtocline</code> : Prevent protrusion (https://tex.stackexchange.com/q/172785/10109) 451	<code>\@setupverbvisibleospace</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 348
2018-09-24 fontdef.dtx v3.0b		General: Start LR-mode if necessary (git/49) 286	<code>\@sverb</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 348
2018-09-24 ltmath.dtx v1.2b		<code>\smash</code> : Start LR-mode if necessary (git/49) 354	<code>\@verbvisibleospacebox</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 348
2018-09-24 ltmath.dtx v1.2b		<code>\vphantom</code> : Start LR-mode if necessary (git/49) 354	<code>\asciispace</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 347
2018-09-24 ltspace.dtx v1.3h		<code>\enspace</code> : Start LR-mode if necessary (git/49) 85	<code>verbatim*</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 349
2018-09-26 ltdefns.dtx v1.5e		<code>\leavevmode@ifvmode</code> : Macro added (git/49) 85	<code>\verbvisibleospace</code> : Provide <code>\verbvisibleospace</code> such that it is usable in normal text (github/70) 347
2018-09-26 ltdefns.dtx v1.5e		<code>\renew@command</code> : Always explicitly generate a space after the <code>csname</code> and not rely on <code>\noexpand</code> to save tokens (git/41) 41	Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69) 347
2018-09-26 ltmiscen.dtx v1.1n		<code>\writefile</code> : Sometimes mask the newline char when writing to files (github/73) 340	2018-10-21 ltluatex.dtx v1.1i
<code>\add@percent@to@temptokena</code> : Sometimes mask the newline char when writing to files (github/73) 339			<code>new_luafunction</code> : Function added 604
			2018-11-09 ltbibl.dtx LaTeX2e
			<code>\bibliography</code> : Zap spaces in the argument as BibTeX doesn't support them (github/88) . . 476
			2018-11-18 ltoutenc.dtx v2.0k
			General: Provide <code>\Hwithstroke</code> and <code>\hwithstroke</code> 137
			2018-11-19 ltoutenc.dtx v2.0k
			General: Added <code>\Hwithstroke</code> and <code>\hwithstroke</code> 121
			2018-11-28 ltoutput.dtx v1.4d
			<code>\@combinedblfloats</code> : Unbox <code>\@outputbox</code> to preserve boxing level (github/94) . . . 513

2018-12-30 lttab.dtx v1.1p	need to apply them inside a group	47
\@tabclassz: Add extra \hskip to guard against an \unskip at the start of a c-column cell (gh/102)		410
2019-02-07 ltfiles.dtx v1.1o		
\@swaptwoargs: Helper macro added		98
\InputIfFileExists: Expand \@filef@und before executing second argument (github/109)		98
2019-02-07 ltffssbas.dtx v3.2b		
\define@newfont: Changed wording of warning (github/107)		164, 187
2019-06-18 ltluatex.dtx v1.1j		
General: finish_synctex_callback renamed finish_synctex		606
font_descriptor_objnum_provider added		606
make_extensible added		606
new_graf added		605
page_objnum_provider added		606
process_pdf_image_content added		606
wrapup_run added		606
2019-07-01 ltfiles.dtx v1.2a		
\IfFileExists: Support UTF-8		97
\includeonly: Support UTF-8		94
\set@curr@file: Support UTF-8		96
2019-07-09 ltffssbas.dtx v3.2c		
\DeclareErrorFont: Don't set any \f@... macros		167, 190
2019-07-09 ltffssini.dtx v3.1c		
General: Explicitly set some defaults		266
2019-08-22 ltxref.dtx v1.1l		
\labelformat: Commanded moved from varioref.sty		335
\Ref: Commanded moved from varioref.sty		335
\refstepcounter: Allow \p@... to have an argument		335
2019-08-27 fontdef.dtx v3.0c		
General: Various commands made robust throughout the file		279
2019-08-27 ltboxes.dtx v1.3b		
General: Various commands made robust		379
2019-08-27 ltdefns.dtx v1.5f		
General: Make various commands robust		52
\MakeRobust: Make the assignments global as we may		
2019-08-27 ltfiles.dtx v1.2b		
\IfFileExists: Make command robust		97
\InputIfFileExists: Make command robust		98
2019-08-27 ltffssbas.dtx v3.2d		
General: Make various commands robust		153, 176
2019-08-27 ltffssdcl.dtx v3.0s		
\@DeclareMathDelimiter: Make math delimiters robust		244
\DeclareMathAccent: Make math accents robust		240
2019-08-27 ltffssini.dtx v3.1d		
General: Make various commands robust		252
2019-08-27 ltidxglo.dtx v1.1f		
General: Make \index and \gloassary robust		472
2019-08-27 ltlength.dtx v1.1d		
General: Make various command robust		152
2019-08-27 ltlogos.dtx v1.1j		
\TeX: Make \TeX command robust		87
2019-08-27 ltmath.dtx v1.2c		
General: Make various commands robust		351
2019-08-27 ltmiscen.dtx v1.1p		
General: Make various commands robust		336
\begin: Make command robust		341
\end: Make command robust		343
2019-08-27 ltoutput.dtx v1.4e		
\@begindvibox: Make \AtBeginDvi robust		493
2019-08-27 ltpage.dtx v1.0l		
General: Make various commands robust		478
2019-08-27 ltpictur.dtx v1.1m		
General: Make various commands robust		414
Remove several unnecessary \gdef definitions		414
2019-08-27 ltsect.dtx v1.1d		
General: Make various commands robust		441
2019-08-27 ltspac.dtx v1.3j		
General: Make various commands robust		71
2019-08-27 lttab.dtx v1.1q		
General: Make various commands robust		392

Remove several unnecessary <code>\gdef</code> definitions	392	<code>\set@curr@file</code> : remove quotes . .	96
2019-08-30 ltterror.dtx v1.2q <code>\conditionally@traceoff</code> : Macro added	68	2019-11-01 ltdirchk.dtx v1.3a <code>\filename@parse</code> : take last . not first	12
<code>\conditionally@traceon</code> : Macro added	68	2019-11-02 ltmiscen.dtx v1.1s <code>\@centercr</code> : Make <code>\@centercr</code> robust (gh/203)	344
2019-09-09 ltfsdcl.dtx v3.0s <code>\DeclareMathSymbol</code> : Allow definition if the math symbol was a command already robust	242	2019-11-02 ltspc.dtx v1.3k <code>\@normalcr</code> : Make also <code>\@normalcr</code> robust	75
2019-09-14 ltfinal.dtx v2.1h <code>\@uclclist</code> : Expand UTF8 chrs when case changing (gh/177)	624	2019-11-09 ltfiles.dtx v1.2e <code>\set@curr@file</code> : expand and <code>\string</code> before removing quotes	96
2019-09-16 ltxref.dtx v1.1m General: Correctly revert the <code>\p@...</code> change	335	2019-11-10 ltmiscen.dtx v1.1r <code>\add@percent@to@temptokena</code> : fix to special comment catcodes (gh/202)	339
2019-09-21 fontdef.dtx v3.0d General: Distangle alias (gh/184)	281, 285	2019-11-11 ltfiles.dtx v1.2f <code>\@iffileonpath</code> : make <code>\@file@und</code> match quoting used on <code>\openin</code>	97
2019-10-02 ltfinal.dtx v2.2 General: Load <code>ltxexpl</code>	626	2019-11-22 ltoutenc.dtx v2.0l General: Avoid spurious if fontenc selects LY1 as default encoding (gh/199)	144
2019-10-02 ltluatex.dtx v1.1k General: <code>linebreak_filter</code> is <code>exclusive</code>	605	2019-11-29 ltclass.dtx v1.3e <code>\@pr@videpackage</code> : Protect package info text (gh/52) . .	561
<code>mlist.to.hlist</code> is <code>exclusive</code> . . .	605	2019-12-17 fontdef.dtx v3.0e <code>\mddefault</code> : Set <code>\bfdefault</code> to “b”	273
<code>process_rule</code> is <code>exclusive</code>	605	<code>\updefault</code> : Set <code>\updefault</code> to “up”	273
2019-10-07 lttab.dtx v1.1q <code>\extracolsep</code> : This needs to expand	403	2019-12-17 ltfntcmd.dtx v3.4c <code>textssc</code> : Macro added	295
2019-10-11 ltfiles.dtx v1.2c <code>\set@curr@file</code> : Remove one brace group	96	2019-12-17 ltfsbas.dtx v3.2e <code>\usefont</code> : Don’t call <code>\fontseries</code> or <code>\fontshape</code>	161, 184
2019-10-11 ltfsstrc.dtx v3.0l <code>\@font@aliasinfo</code> : Added ‘alias’ size function	220	2019-12-17 ltssini.dtx v3.1e General: Provide custom series settings a la mweights	253
2019-10-18 ltclass.dtx v1.3d <code>\@fileswithoptions</code> : Initialize <code>\...-h@ok</code> only when loading the package or class (gh/198)	568	<code>\DeclareEmphSequence</code> : Provide <code>\emph</code> sequences	262
2019-10-22 ltluatex.dtx v1.1j General: <code>page_objnum_provider</code> and <code>process_pdf_image_content</code> classified data	606	2019-12-18 ltoutenc.dtx v2.0m General: Don’t fake <code>\textcompwordmark</code> ; take default from T1 instead	116
2019-10-25 ltmiscen.dtx v1.1q <code>\add@percent@to@temptokena</code> : Allow unbalanced conditionals in #1 (gh/202)	339	<code>\add@accent</code> : Avoid code that breaks <code>\accent</code>	109
2019-10-26 ltfiles.dtx v1.2d <code>\@iffileonpath</code> : quote on <code>openin</code>	97	2019-12-21 fontdef.dtx v3.0e General: Distangle alias (gh/184)	279–281
<code>\IfFileExists</code> : dont quote name	97		
<code>\IfFileExists@</code> : quote on <code>openin</code>	97		

2020-01-11 ltffssini.dtx v3.1f

\rmfamily: Streamlined

implementation with hook . . 257

\ttfamily: Streamlined

implementation with hook . . 258

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

$\backslash!$	b353, b355, <u>B201</u> , Q419	1608, 1609, 1635, 1742, 1812, 1819, 11185, 11267, 11268, 11277, 11278, 11287, 11288, 11305, 11366, 11367, 11397, 11398, o344, p344, x190, x207
$\backslash"$	l226, l378, l416, l454, l465, l573, l605, l632, l640, l646, l650, l656, l660, l666, l672, l679, l680, l686, l690, l740, l786, l1186, l1204, l1210, l1214, l1220, l1224, l1230, l1236, l1243, l1244, l1250, l1254, l1256, l1363, o351, p351, x189, x206, Q420	$\backslash/$ a97, d10, o292, o346, p292, p346, N192
$\backslash\#$	a62, a75, b6, b14, b420, d387, o338, p338, Q403	$\backslash:$ b354, b356, d381, d382, <u>B206</u>
$\backslash\$$ a74, b4, b13, d386, l307, l441, l448, l562, l798, l805, x551, x558, Q404		$\backslash;$ b354, b356, u492, B185, <u>B201</u>
$\backslash\%$ a75, a105, a107, a127, b14, b418, d387, l488, l490, o340, p340, x560, x674, A56, N791, N792, Q405		$\backslash<$ l569, l733, o341, p341, A351, E59, E97
$\backslash\&$ a74, b5, b13, b419, d386, N193, Q406		$\backslash=$ l229, l381, l470, l743, l1183, l1257, l1258, l1273, l1274, l1296, l1297, l1298, l1323, l1324, l1349, l1350, l1379, l1380, l1395, l1396, l1403, l1404, t370, x193, x210, D245, D266, E59
\backslash'	b440, l227, l379, l418, l452, l462, l575, l585, l591, l593, l596, l598, l606, l612, l618, l620, l623, l625, l633, l637, l644, l648, l653, l658, l661, l663, l670, l675, l676, l683, l688, l691, l741, l788, l807, l809, l810, l811, l814, l816, l817, l818, l820, l821, l1180, l1201, l1208, l1212, l1217, l1222, l1225, l1227, l1234, l1239, l1240, l1247, l1252, l1255, l1263, l1264, l1311, l1312, l1317, l1318, l1329, l1330, l1335, l1336, l1364, l1365, l1389, l1390, o350, p350, t370, u245, x184, x201, A351, B208, D245, D266, E60, M584, Q421	$\backslash>$ l566, l734, o342, p342, A351, <u>B201</u> , B206, E59
$\backslash($	<u>B225</u> , B299	$\backslash?$ b353, b355, Q421
$\backslash)$	b440, <u>B225</u> , B300	$\backslash@$ a65, d392, d393, g19, <u>i338</u> , j2, N24, N32, P18, P783, Q413
$\backslash*$ o343, p343, <u>B205</u> , N602, N717, N793		$\backslash@@$ a331, a332, f15, f19, f20, f21, f22, f24, f27, f28, f30, f31, k389, k405, q464, q466, q467, E214, E215, E216, E226, M10, M11
$\backslash+$	E60	$\backslash@@defaultsubs$ <u>o520</u> , <u>p520</u>
$\backslash,$	b354, b356, <u>i335</u> , u498, A351, B7, B8, B40, B154, B156, B159, B184, B201	$\backslash@@enc@update$ l183, o223, <u>o227</u> , p223, <u>p227</u>
$\backslash-$	b322, d9, d394, i326, l413, l414, l568, l782, l783, o345, p345, A351, D244, D265, E60, Q157, Q197	$\backslash@@end$ a69, a222, d8, k369, k370, A39, A86, O18, Q500, Q521
$\backslash.$	b353, b355, k43, k101, l228, l380, l449, l450, l471, l581, l582,	$\backslash@@endpbox$ E181, E212, <u>E362</u>
		$\backslash@@eqnocr$ B334, B352, <u>B355</u> , B460
		$\backslash@@fileswith@pti@ns$ N266, <u>N489</u>
		$\backslash@@hyph$ <u>d9</u>
		$\backslash@@hyphenation$ <u>l205</u>
		$\backslash@@if@newlist$ M599, M644, M657, M703
		$\backslash@@if@definable$ <u>d107</u> , l17
		$\backslash@@input$ a68, d7, k319, k327, k332, k358, A19, N944
		$\backslash@@italiccorr$ <u>d10</u> , w113, w117
		$\backslash@@line$ <u>D402</u>
		$\backslash@@math@bgroup$ w131, w138
		$\backslash@@math@egroup$ <u>w128</u> , <u>w128</u>
		$\backslash@@par$ 69, <u>d6</u> , h4, A86, A222, A227, A230, A244, A248, A251, C82, C85, D221, D243, D264, E187, H67, H118, M257
		$\backslash@@patterns$ <u>l205</u>
		$\backslash@@protect$ d228, d234, d243

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \@startpbox [E181](#), [E212](#), [E362](#)
- \@underline [D358](#), [D361](#), [D362](#)
- \@unprocessedoptions ... [N471](#), [N533](#)
- \@warning [g166](#)
- \@Alph [m106](#), [m122](#)
- \@DeclareEncodingSubset
 . [x66](#), [x68](#), [x69](#), [x70](#), [x71](#), [x74](#), [x81](#)
- \@DeclareMathDelimiter ... [s752](#), [s771](#)
- \@DeclareMathSizes
 [o170](#), [o171](#), [o173](#), [p170](#), [p171](#), [p173](#)
- \@Esphack [i164](#), [I201](#), [I223](#), [I241](#)
- \@IncludeInRelease [c79](#), [c80](#)
- \@IncludeInRelease [c77](#), [c78](#), [c79](#)
- \@M [b21](#), [b397](#), [b398](#), [d22](#), [d24](#),
 [i11](#), [i12](#), [i13](#), [i14](#), [i15](#), [i16](#), [i17](#),
 [i18](#), [i93](#), [o589](#), [o596](#), [p589](#), [p596](#),
 [q393](#), [q406](#), [B344](#), [C194](#), [E56](#),
 [H67](#), [H100](#), [H118](#), [H130](#), [H188](#),
 [H211](#), [M176](#), [M195](#), [M198](#), [M258](#)
- \@MM [b21](#), [M299](#), [I412](#)
- \@Mi [e3](#), [M136](#)
- \@Mii ... [e3](#), [M295](#), [M1159](#), [M1326](#),
 [I53](#), [I122](#), [I194](#), [I216](#), [I241](#), [I311](#)
- \@Miii [e3](#), [M298](#), [I55](#), [I124](#), [I313](#)
- \@Miv .. [e3](#), [M272](#), [I195](#), [I201](#), [I217](#), [I223](#)
- \@Roman [m104](#), [m110](#)
- \@TeXversion [1](#), [6](#), [a326](#), [g28](#)
- \@acci [t370](#), [D245](#), [D266](#)
- \@accii [t370](#), [D245](#), [D266](#)
- \@acciii [t370](#), [D245](#), [D266](#)
- \@acol [E156](#),
 [E166](#), [E236](#), [E237](#), [E249](#), [E250](#),
 [E253](#), [E270](#), [E285](#), [E293](#), [E303](#)
- \@acolampacol [E234](#), [E251](#),
 [E253](#), [E260](#), [E268](#), [E302](#), [E305](#)
- \@activechar@info [M575](#)
- \@addamp [E227](#), [E236](#),
 [E237](#), [E252](#), [E266](#), [E303](#), [E304](#)
- \@addfield [E43](#),
 [E53](#), [E74](#), [E81](#), [E113](#), [E128](#), [E130](#)
- \@addmarginpar [M331](#), [M1811](#)
- \@addtobot [M975](#), [M1062](#),
 [M1129](#), [M1181](#), [M1290](#), [M1349](#)
- \@addtocurcol .. [M328](#), [M1066](#), [M1965](#)
- \@addtodblcol [M854](#), [M1562](#)
- \@addtofilelist
 . [a101](#), [a103](#), [k61](#), [k116](#), [k319](#),
 [k327](#), [k386](#), [t346](#), [t349](#), [t356](#),
 [t359](#), [t366](#), [t369](#), [Q210](#), [Q213](#), [Q542](#)
- \@addtonextcol . [M853](#), [M1386](#), [M1966](#)
- \@addtopreamble [E287](#), [E300](#),
 [E306](#), [E307](#), [E308](#), [E310](#), [E322](#)
- \@addtoreset [m16](#), [m39](#), [m44](#), [m79](#), [m82](#)
- \@addtotoporbot [M1012](#),
 [M1175](#), [M1343](#), [M1435](#), [M1524](#)
- \@afterheading [H92](#), [H125](#)
- \@afterindentfalse [H45](#)
- \@afterindenttrue
 [H43](#), [H124](#), [H187](#), [H210](#)
- \@alph [m105](#), [m118](#), [I375](#)
- \@ampacol ... [E234](#), [E251](#), [E262](#), [E305](#)
- \@arabic
 [m43](#), [m76](#), [m87](#), [m102](#), [m108](#), [I373](#)
- \@argarraycr [E191](#), [E192](#)
- \@argdef [d55](#)
- \@argsbox [D382](#)
- \@argtabularcr [E198](#), [E199](#)
- \@array [E169](#), [E170](#)
- \@arrayacol [E156](#), [E234](#)
- \@arrayclassiv [E157](#), [E307](#)
- \@arrayclassz [E156](#), [E251](#)
- \@arraycr [E158](#), [E189](#), [E191](#)
- \@arrayparboxrestore [D235](#), [D277](#), [E360](#)
- \@arrayrule [E285](#),
 [E287](#), [E291](#), [E293](#), [E295](#), [E322](#)
- \@arstrut [E180](#), [E213](#), [E319](#)
- \@arstrutbox . [E173](#), [E206](#), [E319](#), [E361](#)
- \@author [H8](#), [H32](#)
- \@auxout [k145](#),
 [k157](#), [k172](#), [k191](#), [k206](#), [z33](#),
 [H175](#), [K7](#), [K8](#), [K19](#), [K29](#), [K37](#), [K43](#)
- \@backslashchar [d193](#),
 [g185](#), [g187](#), [u258](#), [N665](#), [N755](#)
- \@badcrerr [g227](#)
- \@badend [g198](#), [A150](#)
- \@badlinearg [g217](#), [F59](#),
 [F68](#), [F69](#), [F73](#), [F117](#), [F122](#), [F133](#)
- \@badmath
 [g201](#), [B229](#), [B231](#), [B236](#), [B239](#),
 [B248](#), [B260](#), [B265](#), [B274](#), [B287](#),
 [B292](#), [B384](#), [B396](#), [B412](#), [B421](#)
- \@badpoptabs [g205](#), [E73](#), [E139](#)
- \@badrequireerror [N206](#), [N541](#)
- \@badtab [g208](#),
 [E22](#), [E75](#), [E96](#), [E102](#), [E109](#), [E136](#)
- \@begin@tempboxa [D27](#),
 [D42](#), [D158](#), [D221](#), [D383](#), [D391](#)
- \@begin@documenthook [k55](#),
 [k58](#), [k110](#), [k113](#), [K33](#), [N500](#), [N514](#)
- \@begin@dv [M623](#), [M682](#), [M710](#)
- \@begin@dvibox [M86](#), [M711](#)
- \@begin@parpenalty
 [i14](#), [B387](#), [B399](#), [B425](#), [C23](#), [C170](#)
- \@begin@theorem [G30](#), [G35](#)
- \@bezier [F369](#), [F370](#)
- \@bibitem [K3](#), [K8](#)
- \@biblabel [K4](#), [K54](#)

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=ltspc.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\@bitor` M15, M881,
 M901, M937, M960, M1027,
 M1111, M1121, M1269, M1280,
 M1422, M1509, M1627, M1752
`\@botlist` M65, M384, M386,
 M431, M433, M717, M738,
 M747, M748, M989, M992,
 M1027, M1029, M1121, M1123,
 M1280, M1282, M1922, M1949
`\@botnum` M109, M986,
 M987, M992, M996, M1003,
 M1458, M1463, M1551, M1558,
 M1914, M1941, M1983, I274
`\@botroom` M110,
 M989, M992, M1915, M1942, I275
`\@boxfpsbit` ... M2031, M2033, M2038
`\@break@tfor` f31, k265, k304, w98
`\@bsphack` i36,
 i99, i286, i302, z32, J6, J18, J23,
 J35, K39, M1882, I52, I121, I310
`\@caption` I12, I14
`\@captype` M1995,
 I5, I9, I12, I40, I88, I109, I157
`\@car` 37, d38, j14, l89, l110
`\@carcube` d40, d110
`\@ccclv` ... b16, M300, M304, M382,
 M383, M412, M429, M430,
 M459, M483, M487, M488, Q53
`\@ccclvi` b21, b57, b82, b93, b95, b99,
 b159, b173, N663, N753, P28, P56
`\@cdr` 37, d38, d362, d363
`\@centercr` .. A153, A179, A183, A187
`\@centering` B314, B315,
 B322, B325, B328, B453, B457
`\@cflb` M714
`\@cflt` M714
`\@changed@cmd` l3, l63,
 l223, o95, o231, p95, p231, Q372
`\@changed@x` l3, l211, l219
`\@changed@x@mouth` l211, l219
`\@charlb` k209, k217
`\@charrb` k211, k217
`\@chclass` E247, E248, E311, E324, E329
`\@check@IncludeInRelease`
 c105, c107, c109
`\@check@c` d164, d166
`\@check@eq` d170, d171, d175
`\@checkend` A11, A137, A149
`\@chnum` E255,
 E274, E311, E326, E327, E328
`\@circ` F341, F342, F343
`\@circle` F329, F330
`\@circlefnt` F37, F39, F233,
 F262, F304, F334, F349, F364
`\@cite` K16, K52
`\@cite@ofmt` K24, K53
`\@citea` K15, K17
`\@citeb` . K16, K18, K19, K20, K23,
 K24, K41, K42, K43, K44, K45
`\@citex` K13, K14
`\@classi` E247, E283
`\@classii` E247, E297
`\@classiii` E247, E302
`\@classiv` E157, E168, E248
`\@classoptionslist` N9, N235, N246,
 N385, N386, N413, N414, N1025
`\@classv` E248, E308
`\@classz` E156, E167, E247
`\@cline` E343
`\@clnht` F75, F76, F84,
 F86, F88, F98, F105, F131, F358
`\@clnwd` . F77, F83, F87, F89, F90, F358
`\@cls@pkg` N145, N146,
 N450, N480, N517, N526, N528,
 N545, N878, N953, N975, N1005
`\@clsextension` ... N16, N41, N52,
 N96, N157, N172, N202, N219,
 N235, N245, N303, N318, N326,
 N384, N412, N484, N492, N518
`\@clubpenalty` k9,
 k23, k80, C128, C196, H106, H135
`\@colht` k20, k77, M114, M231,
 M242, M251, M252, M387,
 M399, M434, M447, M474,
 M505, M535, M541, M545,
 M555, M560, M645, M704,
 M777, M815, M859, M884,
 M903, M943, M965, M1642,
 M1768, M2096, Q88, I273,
 I275, I278, I284, I285, I298, I299
`\@colnum` M111, M995,
 M1040, M1109, M1110, M1138,
 M1146, M1188, M1267, M1268,
 M1300, M1312, M1356, M1420,
 M1421, M1458, M1463, M1507,
 M1508, M1550, M1557, M1910,
 M1937, M1976, M2151, I276
`\@colroom`
 k21, k78, M115, M252, M273,
 M274, M285, M288, M387,
 M434, M777, M994, M1039,
 M1105, M1108, M1137, M1262,
 M1266, M1299, M1416, M1419,
 M1502, M1506, M1911, M1938,
 M2106, M2111, M2156, Q87
`\@combinedblfloats`
 M750, M2230, M2269
`\@combinefloats` M501, M714

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\comdblfilelt` [M750](#)
`\comfleft` [M720](#), [M736](#), [M750](#)
`\cons` ... [37](#), [b196](#), [b213](#), [d37](#), [m44](#),
[M237](#), [M888](#), [M907](#), [M923](#),
[M947](#), [M949](#), [M969](#), [M971](#),
[M1141](#), [M1209](#), [M1305](#), [M1378](#),
[M1451](#), [M1541](#), [M1644](#), [M1667](#),
[M1770](#), [M1795](#), [M1812](#), [M1813](#),
[M2157](#), [I193](#), [I215](#), [I239](#), [I355](#)
`\contfield` [E50](#), [E129](#), [E141](#)
`\ctrerr` [g194](#), [m121](#), [m125](#), [m139](#), [m147](#)
`\curfield` [E16](#), [E41](#),
[E47](#), [E51](#), [E52](#), [E54](#), [E118](#), [E119](#)
`\curline`
[E16](#), [E27](#), [E39](#), [E44](#), [E53](#), [E54](#),
[E55](#), [E78](#), [E79](#), [E91](#), [E116](#), [E117](#)
`\curr@enc` [I154](#), [I156](#)
`\curr@file` [k153](#),
[k161](#), [k222](#), [k241](#), [k339](#), [N608](#),
[N613](#), [N619](#), [N642](#), [N651](#), [N677](#)
`\currbox` [b264](#),
[b265](#), [b266](#), [M213](#), [M214](#), [M225](#),
[M226](#), [M228](#), [M229](#), [M237](#),
[M311](#), [M312](#), [M853](#), [M854](#),
[M1102](#), [M1104](#), [M1112](#), [M1135](#),
[M1139](#), [M1141](#), [M1156](#), [M1197](#),
[M1209](#), [M1257](#), [M1260](#), [M1297](#),
[M1302](#), [M1305](#), [M1322](#), [M1367](#),
[M1378](#), [M1410](#), [M1426](#), [M1440](#),
[M1451](#), [M1493](#), [M1530](#), [M1541](#),
[M1581](#), [M1585](#), [M1596](#), [M1602](#),
[M1604](#), [M1608](#), [M1613](#), [M1622](#),
[M1631](#), [M1637](#), [M1644](#), [M1667](#),
[M1702](#), [M1706](#), [M1718](#), [M1725](#),
[M1727](#), [M1731](#), [M1737](#), [M1747](#),
[M1762](#), [M1770](#), [M1795](#), [M1813](#),
[M1822](#), [M2001](#), [M2002](#), [M2031](#),
[M2061](#), [M2066](#), [M2112](#), [M2115](#),
[M2127](#), [M2135](#), [M2152](#), [M2157](#),
[I60](#), [I91](#), [I95](#), [I129](#), [I160](#), [I164](#),
[I193](#), [I214](#), [I215](#), [I239](#), [I257](#),
[I259](#), [I261](#), [I319](#), [I322](#), [I327](#), [I331](#)
`\currdir` .. [1](#), [6](#), [a108](#), [a130](#), [a132](#),
[a138](#), [a140](#), [a146](#), [a148](#), [a153](#),
[a155](#), [a165](#), [a178](#), [a243](#), [a256](#),
[a269](#), [N591](#), [N613](#), [N642](#), [N726](#)
`\current@cmd` [I25](#), [o235](#), [p235](#)
`\currentlabel` [z34](#), [z41](#),
[z53](#), [z61](#), [B318](#), [B438](#), [D330](#), [I414](#)
`\currenvir` [g199](#), [A3](#),
[A96](#), [A113](#), [A150](#), [C112](#), [D104](#),
[N653](#), [N665](#), [N673](#), [N677](#), [N683](#),
[N743](#), [N755](#), [N763](#), [N767](#), [N773](#)
`\currenvline`
... [g199](#), [A97](#), [A114](#), [A151](#), [D105](#)
`\current` [N15](#), [N23](#), [N31](#),
[N156](#), [N157](#), [N171](#), [N172](#), [N219](#),
[N228](#), [N235](#), [N245](#), [N313](#), [N322](#),
[N442](#), [N447](#), [N448](#), [N453](#), [N459](#),
[N463](#), [N465](#), [N466](#), [N468](#), [N470](#),
[N472](#), [N473](#), [N476](#), [N482](#), [N484](#),
[N492](#), [N510](#), [N518](#), [N534](#), [N535](#)
`\currlist` [M67](#),
[M311](#), [M388](#), [M391](#), [M435](#),
[M438](#), [M1812](#), [I193](#), [I215](#), [I355](#)
`\currname` ... [c66](#), [c82](#), [c90](#), [k397](#),
[k398](#), [N14](#), [N22](#), [N30](#), [N143](#),
[N145](#), [N156](#), [N171](#), [N228](#), [N322](#),
[N441](#), [N465](#), [N466](#), [N468](#), [N470](#),
[N472](#), [N473](#), [N510](#), [N526](#), [N528](#),
[N535](#), [N545](#), [N953](#), [N975](#), [N1005](#)
`\currnamestack` [N20](#)
`\curroptions`
... [N228](#), [N236](#), [N258](#), [N535](#), [N536](#)
`\currsize` [t297](#)
`\currtype` [M119](#), [M878](#),
[M879](#), [M880](#), [M881](#), [M898](#),
[M899](#), [M900](#), [M901](#), [M1027](#),
[M1111](#), [M1121](#), [M1269](#), [M1280](#),
[M1422](#), [M1509](#), [M1627](#), [M1752](#),
[M2001](#), [M2003](#), [M2004](#), [M2007](#)
`\curtab` [E11](#),
[E26](#), [E74](#), [E75](#), [E76](#), [E82](#), [E83](#),
[E86](#), [E90](#), [E91](#), [E95](#), [E134](#), [E135](#)
`\curtabmar` [E11](#), [E25](#),
[E26](#), [E38](#), [E44](#), [E77](#), [E90](#), [E94](#), [E95](#)
`\@d@r` [a161](#), [a162](#)
`\@dashbox` [F176](#), [F177](#),
[F178](#), [F179](#), [F180](#), [F183](#), [F186](#),
[F188](#), [F197](#), [F199](#), [F200](#), [F201](#),
[F202](#), [F205](#), [F208](#), [F211](#), [F360](#)
`\@dashcnt` [F170](#), [F171](#),
[F172](#), [F173](#), [F174](#), [F175](#), [F185](#),
[F187](#), [F190](#), [F191](#), [F192](#), [F193](#),
[F195](#), [F196](#), [F207](#), [F210](#), [F360](#)
`\@dashdim` [F169](#), [F170](#), [F171](#),
[F172](#), [F174](#), [F177](#), [F179](#), [F180](#),
[F181](#), [F185](#), [F187](#), [F189](#), [F190](#),
[F191](#), [F192](#), [F195](#), [F199](#), [F201](#),
[F202](#), [F203](#), [F209](#), [F212](#), [F360](#)
`\@date` [H9](#), [H33](#)
`\@dbflt` [I32](#), [I264](#)
`\@dblarg` [37](#), [d384](#), [H54](#), [H142](#), [I12](#)
`\@dbldeferlist` [M70](#), [M445](#),
[M450](#), [M452](#), [M816](#), [M823](#),
[M824](#), [M1752](#), [M1755](#), [M1795](#),
[M1797](#), [M1926](#), [M1954](#), [I239](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \@dblfloat [I31](#)
- \@dblfloatplacement [k29](#),
[k87](#), [M401](#), [M449](#), [M1907](#),
[M1934](#), [M2235](#), [M2275](#), [I280](#)
- \@dblflset [I26](#)
- \@dblfpbot [M2319](#), [I290](#), [I304](#)
- \@dblfpsep [M2319](#), [I289](#), [I303](#)
- \@dblfpstop [M2319](#), [I288](#), [I302](#)
- \@dbltoplist [M69](#), [M232](#),
[M235](#), [M237](#), [M397](#), [M398](#),
[M445](#), [M446](#), [M755](#), [M759](#),
[M761](#), [M762](#), [M1639](#), [M1644](#),
[M1764](#), [M1770](#), [M1925](#), [M1952](#)
- \@dbltopnum [M107](#), [M127](#),
[M238](#), [M240](#), [M766](#), [M1578](#),
[M1579](#), [M1643](#), [M1646](#), [M1654](#),
[M1674](#), [M1679](#), [M1699](#), [M1700](#),
[M1769](#), [M1773](#), [M1781](#), [M1802](#),
[M1807](#), [M1918](#), [M1945](#), [I283](#), [I297](#)
- \@dbltoproom [M108](#), [M1581](#),
[M1584](#), [M1585](#), [M1594](#), [M1595](#),
[M1598](#), [M1601](#), [M1604](#), [M1608](#),
[M1612](#), [M1616](#), [M1621](#), [M1641](#),
[M1702](#), [M1705](#), [M1706](#), [M1715](#),
[M1716](#), [M1717](#), [M1720](#), [M1724](#),
[M1727](#), [M1731](#), [M1736](#), [M1740](#),
[M1745](#), [M1746](#), [M1767](#), [M1919](#),
[M1946](#), [I284](#), [I286](#), [I298](#), [I300](#)
- \@dec@text@cmd [I3](#)
- \@declaredoptions
[N8](#), [N209](#), [N232](#), [N248](#), [N263](#), [N498](#)
- \@declareoption ... [N207](#), [N208](#), [N216](#)
- \@defaultsubs [o474](#),
[o508](#), [o520](#), [p474](#), [p508](#), [p520](#), [A26](#)
- \@defaultunits .. [o178](#), [o182](#), [o183](#),
[o184](#), [o199](#), [o263](#), [p178](#), [p182](#),
[p183](#), [p184](#), [p199](#), [p263](#), [q133](#), [q135](#)
- \@defdefault@ds ... [N207](#), [N212](#), [N217](#)
- \@deferlist [M68](#), [M384](#), [M393](#),
[M394](#), [M397](#), [M402](#), [M404](#),
[M410](#), [M431](#), [M440](#), [M442](#),
[M778](#), [M786](#), [M787](#), [M798](#),
[M803](#), [M804](#), [M1111](#), [M1114](#),
[M1209](#), [M1211](#), [M1269](#), [M1272](#),
[M1378](#), [M1380](#), [M1422](#), [M1424](#),
[M1451](#), [M1453](#), [M1509](#), [M1511](#),
[M1541](#), [M1543](#), [M1627](#), [M1629](#),
[M1667](#), [M1669](#), [M1924](#), [M1951](#)
- \@definecounter
[m12](#), [m36](#), [B303](#), [C227](#), [C228](#),
[C229](#), [C230](#), [G8](#), [G16](#), [I372](#), [I374](#)
- \@depth [d11](#), [q145](#),
[u550](#), [u551](#), [u553](#), [u554](#), [D357](#),
[D401](#), [E175](#), [E207](#), [F107](#), [F158](#),
[F161](#), [F176](#), [F183](#), [F403](#), [M1851](#)
- \@dir [a160](#), [a163](#), [a165](#), [a167](#), [a168](#)
- \@dischyph [d394](#), [d409](#), [d414](#), [D244](#), [D265](#)
- \@doclearpage [M296](#), [M371](#)
- \@documentclasshook . [N3](#), [N389](#), [N417](#)
- \@doendpe [A138](#), [C123](#)
- \@doilelist [k395](#), [k411](#), [A21](#)
- \@donoparitem [C144](#), [C158](#)
- \@dot [F329](#), [F342](#)
- \@dotsep [H194](#), [H217](#)
- \@dottedtocline ... [H179](#), [H205](#), [H206](#)
- \@downline [F155](#), [F159](#), [F164](#)
- \@downvector [F126](#), [F164](#)
- \@eha [d252](#), [d279](#),
[g170](#), [g188](#), [g190](#), [g192](#), [g200](#),
[g202](#), [g232](#), [k158](#), [k173](#), [l52](#),
[l84](#), [o24](#), [o66](#), [o108](#), [o151](#), [o217](#),
[o274](#), [p24](#), [p66](#), [p108](#), [p151](#),
[p217](#), [p274](#), [q106](#), [s25](#), [s70](#), [s99](#),
[s161](#), [s192](#), [s293](#), [s314](#), [s346](#),
[s387](#), [s432](#), [s437](#), [s492](#), [s610](#),
[s614](#), [s618](#), [s653](#), [s657](#), [s661](#),
[s718](#), [s728](#), [s819](#), [s824](#), [s827](#),
[s876](#), [s881](#), [s884](#), [s918](#), [s921](#),
[s993](#), [s996](#), [s999](#), [s1066](#), [s1072](#),
[w146](#), [x24](#), [x102](#), [x828](#), [x837](#),
[A95](#), [A112](#), [K47](#), [M1876](#), [M1892](#)
- \@ehb [g170](#), [g195](#), [g220](#),
[g222](#), [g224](#), [M234](#), [M390](#), [M437](#)
- \@ehc [d103](#), [d130](#), [g170](#), [g227](#),
[g230](#), [g236](#), [g238](#), [A288](#), [A314](#),
[A327](#), [B359](#), [C220](#), [H31](#), [N1005](#)
- \@ehd . [g170](#), [g197](#), [g204](#), [g207](#), [g209](#),
[g215](#), [s118](#), [E88](#), [E97](#), [N351](#), [I6](#)
- \@elt [d37](#), [k210](#), [m20](#), [m35](#), [m53](#), [m56](#),
[t119](#), [t124](#), [t126](#), [t139](#), [M8](#), [M11](#),
[M15](#), [M27](#), [M30](#), [M31](#), [M32](#),
[M33](#), [M38](#), [M39](#), [M40](#), [M41](#),
[M42](#), [M43](#), [M44](#), [M45](#), [M47](#),
[M51](#), [M57](#), [M58](#), [M59](#), [M60](#),
[M498](#), [M720](#), [M731](#), [M736](#),
[M746](#), [M758](#), [M760](#), [M788](#),
[M805](#), [M825](#), [M844](#), [M857](#),
[M864](#), [M915](#), [M918](#), [M927](#), [M1898](#)
- \@empty [f14](#)
- \@emptycol
[M198](#), [M245](#), [M248](#), [M277](#), [M281](#)
- \@end@check@IncludeInRelease ...
..... [c106](#), [c108](#)
- \@end@tempboxa [D36](#),
[D45](#), [D163](#), [D234](#), [D389](#), [D399](#)
- \@enddocumenthook .. [A10](#), [N500](#), [N515](#)
- \@endfloatbox ... [I190](#), [I211](#), [I236](#), [I248](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\endparenv` C120, C123
`\endparpenalty`
 i15, B388, B400, B426, C23, C124
`\endpbox` E181,
 E212, E242, E309, E360, E363
`\endpefalse` ... A100, A117, C129,
 C131, C135, C136, C138, D107
`\endpeltrue` C138
`\endpettrue` C124, C126, C134
`\endtheorem` G13, G19, G25, G35
`\enlargepage` . M1861, M1866, M1868
`\ensuredmath` B370, B372
`\enumctr` C234, C237, C238
`\enumdepth` .. C226, C232, C233, C234
`\eqcnt` B311, B356,
 B361, B440, B455, B456, B458
`\eqncr` . B323, B341, B362, B363, B442
`\eqnnum` B305, B306, B360, B374, B433
`\eqnsel` B311, B454
`\eqnswfalse` B340
`\eqnswtrue` .. B313, B319, B361, B439
`\eqpen` B311, B344, B346, B353
`\err@` g37,
 g41, g44, g52, g64, g68, g71, g79
`\esphack` i38, i105, i291, i308, z35,
 J17, J19, J34, K50, M1884, I361
`\evenfoot` ... L12, L15, M613, M672
`\evenhead` ... L12, L15, M612, M671
`\expandtwoargs`
 .. d191, N100, N234, N248, N272
`\expast` E215, E243
`\failedlist`
 .. M842, M865, M881, M888,
 M901, M907, M923, M937, M960
`\fcolmadefalse` M833
`\fcolmadetrue` M921
`\filef@und` k252,
 k262, k290, k301, k319, k327, k358
`\filelist`
 . k60, k115, k385, k386, k397,
 t346, t356, t366, Q210, Q526, Q542
`\fileswfalse` k128, N587, N588
`\fileswith@pti@ns` N206,
 N266, N376, N377, N381,
 N383, N410, N411, N438, N489
`\fileswith@ptions`
 N371, N372, N374, N378
`\fileswithoptions`
 N303, N310, N318, N369
`\fileswtrue`
 k7, N571, N574, N628, N632
`\finalstrut` .. D334, D400, E361, I419
`\firstampfalse` ... E230, E253, E270
`\firstamptrue` E238
`\firstcolfirstmark`
 M2212, M2213, M2217
`\firstcoltopmark` ... M2210, M2218
`\firstcolumnfalse` ... M2202, M2247
`\firstcolumntrue` k26,
 k84, M98, M207, M2221, M2253
`\firstofone` d186,
 k54, k109, k233, l68, l153,
 q300, s53, s81, s142, s172, s763,
 A9, B368, E348, K18, K42,
 P65, P100, P108, P166, Q255, I10
`\firstoftwo` a87, d186,
 d336, d346, d356, d383, k263,
 k302, l133, m157, m162, s767,
 x90, x749, x800, x816, z19, L16,
 N48, N68, N80, N107, N125, N1014
`\firsttab` E2, E62, E63, E64, E94, E106
`\flcheckspace` . M989, M1025, M2102
`\flfail` M865,
 M916, M937, M947, M960, M969
`\float` I26, I32
`\floatboxreset` I101, I170, I174
`\floatpenalty`
 .. I3, I53, I55, I58, I122, I124,
 I127, I191, I194, I199, I201,
 I212, I216, I221, I223, I237,
 I241, I311, I313, I317, I321, I355
`\floatplacement`
 k29, k87, M149, M209,
 M253, M477, M1908, M1935, I271
`\flsetnum` M986,
 M1022, M1109, M1267, M1420,
 M1507, M1578, M1699, M2070
`\flsettextmin` M1085,
 M1237, M1406, M1489, M2086
`\flstop` M1972
`\flsucceed`
 M858, M866, M915, M949, M971
`\fltovf` g223, I93, I162, I322
`\flupdates` M992, M1037, M2148
`\flushglue` e17, A180, A183, A188,
 A221, A243, C76, D251, D272
`\fnsymbol` m107, m126
`\font@aliasinfo` q493
`\font@info` o97,
 o135, o141, o301, o318, o556,
 p97, p135, p141, p301, p318,
 p556, q30, q38, q46, q74, q87,
 q126, q154, q168, q179, q193,
 q209, q215, q228, q235, q242,
 q247, q257, q269, q281, q445,
 q457, q462, q469, q499, q512,
 q520, s202, s217, s251, s297,
 s366, s372, s416, s429, s512,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfsscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

s601, s644, s709, s809, s866,
 s1034, s1063, x76, x77, x120, Q374
 \@font@warning o3, o470, o475, o502,
 o509, p3, p470, p475, p502,
 p509, q19, q33, q41, q49, q61,
 q77, q430, q444, q456, q461,
 q468, q511, q519, r30, A23, Q230
 \@fontswitch w126, w128
 \@footnotemark
 I401, I407, I425, I431, I432
 \@footnotetext
 D304, I401, I407, I408, I441, I447
 \@for ... f16, k187, k397, K16, K41,
 N109, N126, N232, N246, N258,
 N263, N283, N293, N536, N578
 \@forloop f19, f20
 \@fornoop f15, f23, f29
 \@fortmp f17, f18, f26, N281, N283
 \@fpbot M863, M2313, I290, I304
 \@fpmin M113, M920, M1916, M1943,
 M2165, M2182, I278, I287, I301
 \@fps M1993,
 M1995, M1998, I41, I42, I44,
 I47, I64, I110, I111, I113, I116, I133
 \@fpsadddfault
 M1990, I45, I48, I114, I117
 \@fpsep M861, M870,
 M942, M964, M2313, I289, I303
 \@fpstype M983,
 M1004, M1005, M1019, M1050,
 M1051, M1075, M1077, M1080,
 M1082, M1133, M1189, M1190,
 M1225, M1228, M1231, M1234,
 M1295, M1357, M1358, M1396,
 M1398, M1401, M1403, M1477,
 M1480, M1483, M1486, M1575,
 M1590, M1592, M1610, M1619,
 M1655, M1656, M1696, M1711,
 M1713, M1733, M1743, M1782,
 M1783, M1986, M2002, M2004,
 M2006, M2009, M2010, M2011,
 M2013, M2014, M2018, M2019,
 M2021, M2022, M2056, M2058,
 M2060, M2072, M2074, M2088,
 M2090, M2120, M2123, M2134
 \@fptop M860, M2313, I288, I302
 \@frameb@x D134, D162, D164
 \@framebox D141, D148, D152
 \@framepicbox D141, D148, D185
 \@freelist b196, b213, b264, M29,
 M34, M48, M56, M213, M499,
 M732, M747, M761, M866,
 M1812, M1813, I60, I129, I319, I320
 \@getcirc F223, F256, F300, F332
 \@getfpsbit M980,
 M1016, M1572, M1693, M2029
 \@getllarrow F124, F132, F134
 \@getlinechar F70, F109
 \@getpen i34, i37, i46, i91
 \@getrarrow F125, F132, F141
 \@glossaryfile J21, J22, J31
 \@gnewline i82, i84, i85
 \@gobble ... d86, d108, d183, d193,
 d210, d214, d249, d255, d258,
 d267, d276, d282, d285, d294,
 d309, d313, d315, f6, f9, g101,
 g127, g153, g162, i75, i381, k61,
 k116, k236, k238, k385, l29,
 o471, o504, p471, p504, q299,
 r26, s28, s30, s255, s266, s330,
 s377, s378, s407, s413, s421,
 s426, s444, s458, s468, s477,
 s490, s507, s516, s590, s592,
 s596, s604, s638, s647, s699,
 s701, s712, s800, s801, s804,
 s812, s859, s869, s951, s956,
 s1025, s1056, t139, t349, t359,
 t369, x777, A104, H143, H144,
 H145, H146, H147, H176, K11,
 K25, K26, M619, M620, M621,
 M678, M679, M680, M927,
 M1900, M2166, M2183, N339,
 N557, N622, N651, N736, N741,
 N815, N959, N971, P64, P98,
 Q213, Q307, Q464, Q473, Q542, I7
 \@gobble@IncludeInRelease
 c86, c94, c103
 \@gobblecr i379, i380
 \@gobblefour d183,
 s24, s252, s368, s370, s374, s376,
 s386, s390, s514, s566, N653, N743
 \@gobbletwo d150,
 d151, d183, f12, k30, k88, o476,
 o510, p476, p510, s132, A16,
 A24, L11, L13, N621, N735, Q236
 \@gtempa d101, d102, d156,
 d158, k366, k367, k369, k370,
 k371, E3, E5, E6, E7, E8, N142,
 N143, N158, N160, N173, N175
 \@halfwidth F2, F38,
 F40, F42, F107, F157, F160,
 F176, F183, F197, F207, F210,
 F366, F388, F401, F402, F403
 \@halignto .. E158, E162, E165, E179
 \@hangfrom H66, H117, H138
 \@height b391, d11, i296,
 i304, l293, l295, q144, u332,
 u550, u551, u553, u554, D118,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- D123, D171, D181, D357, D401,
 E174, E207, E335, E352, F107,
 F158, F161, F176, F183, F199,
 F206, F281, F324, F402, M1851
 \@highpenalty i92, [Q3](#)
 \@hightab [E11](#), [E21](#), [E23](#), [E62](#),
 [E74](#), [E83](#), [E84](#), [E99](#), [E134](#), [E135](#)
 \@hline [F61](#), [F106](#), [F123](#)
 \@holdpg [M122](#), [M300](#),
 [M302](#), [M303](#), [M308](#), [M309](#), [M310](#)
 \@hspace [i350](#), [i351](#)
 \@hspacer [i350](#), [i352](#)
 \@hvector [F119](#), [F123](#)
 \@icentercr [A170](#), [A171](#)
 \@iden [d189](#)
 \@if [d146](#), [d147](#), [d149](#)
 \@if@ptions [N100](#),
 [N104](#), [N106](#), [N123](#), [N124](#), [N140](#)
 \@if@ptions [N95](#), [N96](#), [N99](#), [N101](#), [N448](#)
 \@ifatmargin [E55](#), [E94](#)
 \@ifbothcounters
 [m61](#), [m69](#), [m71](#), [m79](#), [m81](#), [m93](#)
 \@ifclasslater [556](#), [N51](#)
 \@ifclassloaded [556](#), [N40](#)
 \@ifclasswith [556](#), [N95](#)
 \@ifdefinable [37](#), [d59](#),
 [d61](#), [d105](#), [d107](#), [d212](#), [l14](#), [l17](#),
 [m11](#), [n3](#), [t293](#), [D70](#), [G7](#), [G15](#), [G22](#)
 \@iffileonpath [k248](#), [k256](#), [k286](#), [k295](#)
 \@ifl@aded . [N40](#), [N41](#), [N44](#), [N50](#), [N447](#)
 \@ifl@t@r [N56](#),
 [N61](#), [N63](#), [N74](#), [N75](#), [N85](#), [N360](#)
 \@ifl@ter [l1503](#),
 [l1504](#), [N51](#), [N52](#), [N55](#), [N58](#), [N476](#)
 \@ifl@ter@@ [l1503](#), [l1504](#)
 \@ifnch [d368](#), [d370](#), [d382](#)
 \@ifnextchar [37](#),
 [a98](#), [d364](#), [d369](#), [d383](#), [i80](#), [i380](#),
 [k332](#), [m13](#), [q365](#), [A169](#), [B309](#),
 [C143](#), [D9](#), [D11](#), [D18](#), [D20](#), [D26](#),
 [D47](#), [D76](#), [D77](#), [D83](#), [D84](#),
 [D91](#), [D95](#), [D140](#), [D141](#), [D147](#),
 [D148](#), [D153](#), [D186](#), [D194](#), [D202](#),
 [D209](#), [D213](#), [D282](#), [D286](#), [D290](#),
 [D341](#), [D346](#), [D369](#), [D376](#), [D381](#),
 [E57](#), [E169](#), [E191](#), [E198](#), [F10](#),
 [F43](#), [F54](#), [F239](#), [G3](#), [G5](#), [G28](#),
 [K3](#), [K13](#), [M209](#), [M1974](#), [N148](#),
 [N355](#), [N370](#), [N375](#), [N572](#), [N575](#),
 [I27](#), [I264](#), [I324](#), [I399](#), [I422](#), [I439](#)
 \@iforloop [f21](#), [f22](#)
 \@ifpackagelater [556](#), [N51](#)
 \@ifpackageloaded .. [556](#), [M1958](#), [N40](#)
 \@ifpackagewith [556](#), [N95](#)
 \@ifframebox [D154](#), [D155](#), [D156](#)
 \@ifframepicbox [D186](#), [D187](#)
 \@ifstar [37](#),
 [d48](#), [d383](#), [i59](#), [i71](#), [i280](#), [i350](#),
 [m67](#), [m77](#), [o170](#), [p170](#), [r121](#),
 [A158](#), [A165](#), [A338](#), [A347](#), [B343](#),
 [E56](#), [E190](#), [E197](#), [F53](#), [F329](#),
 [H52](#), [H142](#), [M1856](#), [N207](#), [N229](#)
 \@ifundefin@d@i [d326](#), [d327](#), [d344](#), [d347](#)
 \@ifundefin@d@ii ... [d326](#), [d329](#), [d332](#)
 \@ifundefined
 .. [37](#), [d102](#), [d109](#), [d129](#), [d136](#),
 [d158](#), [d169](#), [d249](#), [d255](#), [d276](#),
 [d282](#), [d309](#), [d321](#), [m3](#), [m7](#), [m16](#),
 [m50](#), [m62](#), [m64](#), [o64](#), [o150](#), [p64](#),
 [p150](#), [q378](#), [s287](#), [x75](#), [x779](#),
 [z23](#), [A62](#), [A79](#), [A94](#), [A111](#), [G21](#),
 [K20](#), [K44](#), [L3](#), [L7](#), [N38](#), [N197](#), [N259](#)
 \@ignorefalse [A4](#), [A99](#), [A116](#), [A139](#), [I360](#)
 \@ignoretrue [i174](#), [i187](#),
 [A4](#), [A7](#), [B302](#), [B305](#), [B337](#), [B463](#)
 \@iiiminipage
 .. [D284](#), [D288](#), [D291](#), [D292](#), [D293](#)
 \@iiiparbox [D196](#), [D204](#),
 [D211](#), [D214](#), [D215](#), [D216](#), [D321](#)
 \@iiiminipage [D287](#), [D289](#)
 \@iininput [k332](#), [k333](#)
 \@iiparbox [D210](#), [D212](#)
 \@iirsbox [D381](#), [D390](#)
 \@imakebox [D26](#), [D41](#), [D93](#)
 \@imakepicbox .. [D47](#), [D48](#), [D98](#), [D188](#)
 \@iminipage [D283](#), [D285](#)
 \@include [k161](#), [k174](#), [k178](#)
 \@includeinreleasefalse
 [c69](#), [c74](#), [c99](#), [c104](#), [d418](#)
 \@includeinreleasetrue [c89](#)
 \@index [J18](#), [J19](#), [J35](#)
 \@indexfile [J4](#), [J5](#), [J14](#)
 \@inlabelfalse
 .. [C28](#), [C104](#), [C184](#), [M163](#), [M190](#)
 \@inlabeltrue [C28](#), [C178](#)
 \@inmatherr ... [g233](#), [C112](#), [C142](#), [F329](#)
 \@inmathwarn [l3](#)
 \@inpenc@test [Q252](#), [Q319](#)
 \@input ... [k32](#), [k90](#), [k181](#), [k357](#), [H152](#)
 \@input@ . [k196](#), [k359](#), [o328](#), [p328](#), [K31](#)
 \@inputcheck
 .. [a70](#), [a191](#), [a192](#), [a195](#), [a203](#),
 [d23](#), [d30](#), [k3](#), [k243](#), [k244](#), [k251](#),
 [k260](#), [k261](#), [k264](#), [k281](#), [k282](#),
 [k289](#), [k299](#), [k300](#), [k303](#), [N610](#),
 [N611](#), [N647](#), [N723](#), [N724](#), [N731](#)
 \@insertfalse [M1073](#), [M1223](#),
 [M1394](#), [M1475](#), [M1570](#), [M1691](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\inserttrue` M999, M1044,
 M1161, M1329, M1649, M1776
`\invalidchar` [g238](#)
`\iparbox` [D195](#), [D203](#), [D208](#)
`\irsbox` [D369](#), [D376](#), [D381](#), [D382](#)
`\isavebox` [D91](#), [D92](#)
`\isavepicbox` [D96](#), [D97](#)
`\ishortstack` [F44](#), [F52](#)
`\istackcr` [F54](#), [F55](#)
`\itabcr` [E57](#), [E58](#)
`\item` [C143](#), [C156](#)
`\itemdepth` .. [C241](#), [C243](#), [C244](#), [C245](#)
`\itemfudge` [E38](#), [E44](#), [E70](#)
`\itemitem` [C245](#), [C248](#)
`\itemlabel` [C44](#), [C96](#), [C143](#)
`\itempenalty` [i16](#), [C23](#), [C175](#)
`\iwhiledim` [f7](#)
`\iwhilenum` [f3](#)
`\iwhiles` [f10](#)
`\ixpt` [o652](#), [p652](#)
`\ixstackcr` [F53](#)
`\killglue` [F22](#), [F30](#), [F36](#)
`\kludgeins` ... [M319](#), [M320](#), [M321](#),
 [M323](#), [M376](#), [M377](#), [M423](#),
 [M424](#), [M502](#), [M518](#), [M519](#),
 [M525](#), [M526](#), [M527](#), [M536](#),
 [M552](#), [M556](#), [M566](#), [M1852](#), [M1883](#)
`\labels` [C27](#),
 [C146](#), [C147](#), [C189](#), [C206](#), [C207](#)
`\largefloatcheck` [I192](#), [I213](#), [I238](#), [I256](#)
`\lastchclass` [E238](#),
 [E248](#), [E249](#), [E251](#), [E259](#), [E284](#),
 [E298](#), [E302](#), [E311](#), [E324](#), [E325](#)
`\latex@error` [d103](#),
 [d130](#), [d250](#), [d277](#), [g136](#), [g168](#),
 [g184](#), [g190](#), [g192](#), [g195](#), [g197](#),
 [g199](#), [g202](#), [g204](#), [g206](#), [g209](#),
 [g213](#), [g218](#), [g222](#), [g224](#), [g226](#),
 [g227](#), [g229](#), [g232](#), [g236](#), [g238](#),
 [k158](#), [k173](#), [l50](#), [l84](#), [o5](#), [o24](#),
 [o66](#), [o108](#), [o151](#), [o217](#), [o274](#),
 [p5](#), [p24](#), [p66](#), [p108](#), [p151](#), [p217](#),
 [p274](#), [q105](#), [r100](#), [r111](#), [s23](#), [s68](#),
 [s97](#), [s117](#), [s159](#), [s190](#), [s213](#), [s229](#),
 [s293](#), [s314](#), [s346](#), [s386](#), [s390](#),
 [s432](#), [s437](#), [s492](#), [s560](#), [s566](#),
 [s610](#), [s614](#), [s618](#), [s653](#), [s657](#),
 [s661](#), [s718](#), [s728](#), [s819](#), [s824](#),
 [s827](#), [s876](#), [s881](#), [s884](#), [s918](#),
 [s921](#), [s993](#), [s996](#), [s999](#), [s1066](#),
 [s1072](#), [t43](#), [t54](#), [t275](#), [t318](#),
 [w143](#), [A95](#), [A112](#), [A288](#), [A314](#),
 [A326](#), [B359](#), [C219](#), [E88](#), [E97](#),
 [H31](#), [K47](#), [N314](#), [N333](#), [N346](#),
 [N449](#), [N525](#), [N542](#), [N550](#), [N555](#),
 [N582](#), [N637](#), [N974](#), [N1004](#), [I6](#), [I83](#)
`\latex@info` [d198](#),
 [d269](#), [d296](#), [d397](#), [g136](#), [l85](#), [x97](#)
`\latex@info@no@line` [g136](#), [M576](#)
`\latex@warning` .. [g136](#), [g166](#), [l55](#),
 [z14](#), [F235](#), [K22](#), [K45](#), [M1996](#),
 [N676](#), [N682](#), [N766](#), [N772](#), [I260](#)
`\latex@warning@no@line`
 .. [d177](#), [g136](#), [g167](#), [k17](#), [k74](#),
 [k383](#), [z8](#), [z26](#), [z27](#), [A31](#), [H32](#),
 [M243](#), [M275](#), [M1827](#), [M2062](#),
 [N144](#), [N361](#), [N477](#), [N612](#), [N618](#),
 [N641](#), [N725](#), [N732](#), [N799](#), [N876](#)
`\latexbug` [g225](#), [M333](#), [M1813](#)
`\latexerr` [g166](#),
 [M234](#), [M390](#), [M437](#), [M1874](#), [M1891](#)
`\lbibitem` [K3](#), [K4](#)
`\ldots` [u496](#), [u498](#)
`\leftcolumn` [M121](#),
 [M2203](#), [M2224](#), [M2248](#), [M2257](#)
`\leftmark` [L16](#), [L50](#)
`\let@token` [d368](#),
 [d371](#), [d374](#), [d382](#), [i320](#), [i321](#),
 [i328](#), [w83](#), [w96](#), [B210](#), [B212](#), [B215](#)
`\lign` [B195](#), [B197](#)
`\linechar` [F70](#),
 [F71](#), [F72](#), [F76](#), [F77](#), [F79](#), [F84](#),
 [F86](#), [F87](#), [F88](#), [F89](#), [F91](#), [F95](#),
 [F96](#), [F99](#), [F100](#), [F105](#), [F130](#), [F356](#)
`\linefnt` [F37](#), [F39](#),
 [F70](#), [F123](#), [F131](#), [F162](#), [F165](#), [F363](#)
`\linelen`
 [F58](#), [F59](#), [F83](#), [F90](#), [F99](#), [F101](#),
 [F106](#), [F107](#), [F108](#), [F116](#), [F117](#),
 [F158](#), [F161](#), [F163](#), [F164](#), [F357](#)
`\listctr` [C202](#), [C225](#), [K9](#)
`\listdepth`
 .. [C23](#), [C35](#), [C38](#), [C43](#), [C99](#), [D305](#)
`\listfiles` [k59](#), [k114](#), [k389](#), [k404](#)
`\loadwithoptions` . [N320](#), [N326](#), [N330](#)
`\lowpenalty` [i91](#), [Q3](#)
`\ltab` [E59](#), [E94](#)
`\m` [b21](#), [b351](#), [b353](#),
 [b354](#), [b387](#), [b388](#), [i238](#), [i342](#),
 [i347](#), [k43](#), [k101](#), [C80](#), [F93](#), [F97](#), [K17](#)
`\mainaux` [k5](#), [k35](#), [k36](#),
 [k93](#), [k94](#), [k145](#), [k181](#), [k206](#), [A15](#)
`\makebox` [D11](#), [D20](#), [D25](#)
`\makecaption` [I24](#)
`\makecol` .. [M261](#), [M413](#), [M460](#), [M480](#)
`\makefcolumn`
 .. [M393](#), [M394](#), [M402](#), [M404](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- M440, M442, M450, M452,
 M2161, M2163, M2179, M2180
 \@makefnmark I376, I435
 \@makefntext D333, I418
 \@makeoother a76,
 a97, a126, d386, d387, o341,
 o342, o343, o344, o345, o346,
 o347, o348, o349, o350, o351,
 p341, p342, p343, p344, p345,
 p346, p347, p348, p349, p350,
 p351, A232, A253, A320, A335,
 A345, N192, N193, N658, N748
 \@makepicbox D10, D19, D46, F212
 \@makespecialcolbox M503, M522
 \@marbox M1812, M1822,
 M1825, M1833, M1835, M1836,
 M1838, M1839, M1840, M1849,
 I320, I322, I326, I330, I331, I355
 \@marginparreset I339, I346
 \@markright L33, L48
 \@maxdepth
 k57, k112, M91, M486, M514, Q85
 \@maxtab E2, E82
 \@medpenalty i92, Q3
 \@meta@family@list ... t119, t125, t208
 \@midlist M66,
 M499, M500, M1027, M1029,
 M1141, M1305, M1923, M1950
 \@minipagefalse C181,
 D278, D280, D318, I187, I250, I341
 \@minipagerestore D306, D308
 \@minipagetrue D279, I186
 \@minus d11, M2306,
 M2307, M2308, M2311, M2312
 \@missingfileerror
 557, k340, k351, k360, N470
 \@mkboth L11, L13
 \@mklab C45, C140
 \@mkpream E177, E210, E238
 \@mparbottom
 . M118, M476, M1823, M1831,
 M1832, M1833, M1834, I363, I364
 \@mpargs D297, D321
 \@mparswitchfalse M102
 \@mpfn D303, I399, I404, I444, I448
 \@mpfootins D312,
 D313, D316, D322, D325, D326
 \@mpfootnotetext D304, D324
 \@mplistdepth D305, D322
 \@multicnt
 E346, E348, E349, E350, E357,
 E358, E359, F30, F31, F33,
 F353, F386, F388, F389, F390,
 F391, F395, F399, F410, F414
 \@multiplelabels
 k31, k89, z25, z31, A29, A35
 \@multiput F28, F29
 \@multispan E347, E351, E355
 \@namedef 37, d35, k216, o99, o100,
 o124, p99, p100, p124, q372,
 x78, x114, x782, z28, A110,
 A136, A294, A303, B363, B364,
 E163, G12, G13, G18, G19,
 G23, G24, G25, N574, Q376, Q377
 \@nameuse 37, d36,
 k204, k215, G23, L5, M607, M665
 \@nbitem C168, C221
 \@ne b16
 \@needsf@rmat N356, N359, N364
 \@needsformat N344, N354, N358
 \@negargfalse F66
 \@negargtrue F65
 \@newcommand d54, d55
 \@newctr m13, m15, G8
 \@newenv d125, d126, d135
 \@newenva d123, d124
 \@newenvb d125, d126
 \@newl@bel z22, A17, K10
 \@newline i81, i83
 \@newlistfalse C29,
 C33, C108, C182, M600, M658
 \@newlisttrue C29, C33, C87
 \@next b264, M9, M213, M311, M877,
 M897, M1812, I60, I129, I319, I320
 \@nextchar E245, E246, E306, E307, E308
 \@nil a161, a162, c13, c19, c83,
 c84, d38, d39, d40, d110, d362,
 d363, f13, f19, f27, j14, l89, l110,
 l978, l982, l1013, l1025, l1027,
 o293, o304, o420, o435, o539,
 o542, o543, o551, p293, p304,
 p420, p435, p539, p542, p543,
 p551, q304, q305, q307, q320,
 q326, q330, q331, q367, q388,
 q393, q473, q487, r26, r44, r53,
 r57, s40, s356, s364, s397, s1077,
 s1079, w58, w62, E343, E344,
 N27, N29, N64, N65, N76, N77,
 N88, N89, N91, N282, N286,
 N292, N296, N424, N433, N849,
 N854, N857, N859, N860, N875,
 N895, N912, N966, N988, N1012
 \@nmblistfalse C33, C46, C91
 \@nmblisttrue C225
 \@nnil f13, f20,
 f21, f22, f28, o178, o182, o183,
 o184, o199, p178, p182, p183,
 p184, p199, q133, q135, q299,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- q301, q313, q315, q320, q334,
 q336, q343, q354, q355, q357,
 q388, q393, N396, N397, N404
 \@no@font@optfalse r17, r129
 \@no@lnbk i9, i10, i40
 \@no@pgbk i7, i8, i32
 \@nobreakfalse i94,
 i96, C193, H94, H129, H157,
 M165, M192, M1150, M1316, I182
 \@nobreaktrue i95, H126, I181
 \@nocnterr g191
 \@nocounterr
g191, m4, m8, m16, m62, m64, G21
 \@nodocument g196, k65, k120,
 A87, M156, M183, M212, I39, I108
 \@noitemargfalse C32, C200
 \@noitemargtrue C32, C143
 \@noitemerr g228,
 i218, i253, i276, C69, C81, C107
 \@noligs A233, A254, A336, A346, A357
 \@nolnerr .. g189, i42, i87, A157, A164
 \@nomath o1, o272, p1, p272,
 t226, t261, t267, t288, t290, t295
 \@noparitemfalse C30, C145
 \@noparitemtrue C30, C66
 \@noparlistfalse C31, C70
 \@noparlisttrue C31, C67
 \@normalcr i52, i79, D277
 \@normalsize N4, N5
 \@noskipsecfalse
 k52, k107, H98, M158, M185
 \@noskipsectrue H38, H95
 \@notdefinable d111, d112, d116, g183
 \@notprerr g231, k63, k118
 \@nthm G3, G4
 \@nxrtabmar E11, E21, E23,
 E25, E63, E99, E100, E106, E107
 \@obsoletefile k382
 \@oddfoot
 L11, L14, L15, M124, M610, M669
 \@oddhead L11, L14, M123, M610, M669
 \@onefilewithoptions
 ... N388, N392, N398, N416,
 N420, N426, N439, N488, N825
 \@onelevel@sanitize .. d388, I42, I111
 \@onlypreamble d41,
 d163, d165, d174, d182, k125,
 k134, k155, k384, k410, l23, l24,
 l61, l62, l66, l125, l145, l189,
 l190, l204, o17, o79, o81, o87,
 o103, o131, o146, o167, o172,
 o214, o447, p17, p79, p81, p87,
 p103, p131, p146, p167, p172,
 p214, p447, q373, r28, r36, r42,
 r79, r83, r88, r93, r98, r108,
 r126, r127, r128, r134, r138,
 r142, s17, s19, s44, s46, s107,
 s116, s136, s243, s244, s247,
 s279, s317, s319, s321, s334,
 s349, s396, s398, s440, s479,
 s495, s572, s666, s675, s731,
 s734, s737, s757, s770, s889,
 s924, s934, s948, s1002, s1022,
 s1026, s1090, w140, w141, x79,
 x80, x81, x783, z30, J12, J29,
 K40, N10, N12, N18, N19,
 N26, N28, N34, N36, N39,
 N42, N43, N50, N53, N54,
 N58, N85, N97, N98, N101,
 N140, N149, N181, N183, N200,
 N203, N204, N215, N216, N217,
 N224, N230, N243, N256, N268,
 N270, N275, N299, N304, N308,
 N311, N319, N324, N327, N331,
 N340, N353, N358, N364, N373,
 N378, N438, N488, N490, N499,
 N512, N513, N516, N523, N532,
 N539, N540, N548, N553, N558,
 N805, N806, N807, N808, N810
 \@opargbegintheorem G32, G35
 \@opcol M262, M270,
 M394, M413, M442, M460, M465
 \@options N269
 \@othm G3, G20
 \@outerparskip
 C8, C88, C117, C152, C222
 \@outputbox M120,
 M483, M485, M505, M508,
 M509, M529, M531, M532,
 M537, M540, M545, M547,
 M554, M560, M562, M636,
 M695, M723, M729, M739,
 M740, M763, M770, M856,
 M859, M862, M868, M869,
 M2203, M2207, M2208, M2222,
 M2228, M2248, M2254, M2263
 \@outputdblcol M468,
 M2198, M2200, M2244, M2245
 \@outputpage
 .. M403, M452, M470, M590,
 M2232, M2237, M2270, M2278
 \@oval F239, F247, F292
 \@ovbtrue F248, F293
 \@ovdx F217, F258, F260, F266, F268,
 F280, F283, F302, F310, F312,
 F323, F325, F375, F376, F377,
 F378, F392, F393, F395, F409

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfsscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \@ovdy [F217](#), [F259](#), [F261](#), [F267](#), [F268](#),
[F273](#), [F278](#), [F303](#), [F311](#), [F312](#),
[F317](#), [F321](#), [F382](#), [F383](#), [F384](#),
[F385](#), [F396](#), [F397](#), [F399](#), [F413](#)
- \@ovhlinefalse [F249](#)
- \@ovhlinetrue
... [F242](#), [F246](#), [F254](#), [F260](#), [F288](#)
- \@ovhorz [F265](#),
[F266](#), [F279](#), [F309](#), [F310](#), [F322](#)
- \@ovltrtrue [F248](#), [F293](#)
- \@ovvri [D33](#), [F217](#), [F257](#),
[F273](#), [F284](#), [F301](#), [F317](#), [F326](#)
- \@ovvro [F217](#), [F257](#), [F266](#), [F267](#), [F272](#),
[F278](#), [F279](#), [F301](#), [F310](#), [F311](#),
[F316](#), [F321](#), [F322](#), [F333](#), [F340](#)
- \@ovrtrue [F248](#), [F293](#)
- \@ovttrue [F248](#), [F293](#)
- \@ovvert [F263](#),
[F264](#), [F270](#), [F305](#), [F307](#), [F314](#)
- \@ovvlinefalse [F249](#)
- \@ovvlinetrue [F245](#), [F253](#), [F261](#)
- \@ovxx [F217](#), [F251](#), [F253](#),
[F254](#), [F258](#), [F264](#), [F265](#), [F279](#),
[F296](#), [F298](#), [F302](#), [F307](#), [F309](#),
[F322](#), [F372](#), [F373](#), [F374](#), [F378](#),
[F387](#), [F388](#), [F394](#), [F395](#), [F408](#)
- \@ovvy [F217](#), [F252](#), [F253](#), [F254](#), [F259](#),
[F266](#), [F270](#), [F297](#), [F298](#), [F303](#),
[F310](#), [F314](#), [F379](#), [F380](#), [F381](#),
[F385](#), [F387](#), [F398](#), [F399](#), [F412](#)
- \@p@pfilename [N27](#), [N29](#), [N34](#)
- \@pagedp [M117](#), [M308](#), [M313](#),
[M1091](#), [M1244](#), [M1841](#), [M1851](#)
- \@pageht [M116](#), [M309](#), [M313](#),
[M315](#), [M316](#), [M317](#), [M321](#),
[M1090](#), [M1243](#), [M1824](#), [M1831](#)
- \@par [h3](#), [h5](#)
- \@parboxrestore [D221](#), [D277](#),
[D302](#), [D329](#), [M219](#), [M601](#),
[M659](#), [I19](#), [I100](#), [I169](#), [I338](#), [I413](#)
- \@parboxto [D216](#)
- \@parmoderr [g221](#), [I58](#), [I127](#), [I316](#)
- \@parse@version
[c83](#), [c84](#), [N70](#), [N76](#), [N77](#), [N88](#),
[N860](#), [N875](#), [N912](#), [N988](#), [N1012](#)
- \@parse@version@ [N64](#), [N65](#), [N70](#), [N82](#)
- \@parse@version@dash [N89](#), [N91](#)
- \@partaux ... [k5](#), [k157](#), [k172](#), [k191](#),
[k193](#), [k194](#), [k200](#), [k209](#), [k211](#), [k214](#)
- \@partlist [k153](#), [k169](#), [k187](#)
- \@partswfalse [k8](#)
- \@partswtrue [k151](#), [k168](#)
- \@pass@options
... [N195](#), [N200](#), [N201](#), [N202](#), [N463](#)
- \@pboxswfalse [D219](#), [D295](#)
- \@pboxswtrue [D229](#)
- \@penup [B186](#), [B187](#)
- \@percentchar
[a106](#), [N650](#), [N652](#), [N654](#), [N656](#),
[N740](#), [N742](#), [N744](#), [N746](#), [N794](#)
- \@picbox [F6](#), [F13](#), [F19](#), [F20](#)
- \@picht [F6](#), [F12](#), [F19](#)
- \@picture [F10](#), [F11](#)
- \@picture@warn [F103](#), [F227](#), [F231](#), [F235](#)
- \@pkgextension [N16](#),
[N40](#), [N51](#), [N95](#), [N201](#), [N310](#),
[N313](#), [N330](#), [N399](#), [N427](#), [N534](#)
- \@plus . [d11](#), [i356](#), [H16](#), [H185](#), [H208](#),
[L54](#), [M2306](#), [M2307](#), [M2308](#),
[M2311](#), [M2312](#), [M2316](#), [M2317](#),
[M2318](#), [M2322](#), [M2323](#), [M2324](#)
- \@pnumwidth [H197](#), [H220](#)
- \@popfilename [N20](#), [N485](#)
- \@pr@videopackage
.. [N148](#), [N150](#), [N168](#), [N170](#), [N181](#)
- \@preamble [E178](#), [E180](#),
[E188](#), [E213](#), [E232](#), [E234](#), [E235](#),
[E239](#), [E254](#), [E272](#), [E273](#), [E310](#)
- \@preamblecmds
... [d41](#), [k64](#), [k119](#), [N1026](#), [N1027](#)
- \@preamerr ... [g210](#), [E187](#), [E250](#), [E331](#)
- \@process@pti@ns
..... [N242](#), [N255](#), [N257](#), [N268](#)
- \@process@ptions .. [N229](#), [N231](#), [N243](#)
- \@protected@testopt [d64](#), [d76](#)
- \@providesfile . [a98](#), [a99](#), [N184](#), [Q538](#)
- \@optionlist [N37](#),
[N100](#), [N228](#), [N453](#), [N459](#), [N535](#)
- \@pushfilename [N20](#), [N440](#)
- \@put [F238](#), [F268](#), [F312](#), [F340](#)
- \@qend [d111](#), [d362](#), [g187](#)
- \@qrelax [d112](#), [d362](#)
- \@rc@ifdefinable [d105](#), [d107](#), [d212](#), [I14](#)
- \@reargdef [d97](#)
- \@refundefined [k53](#), [k108](#), [z3](#), [A27](#)
- \@reinserts [M327](#), [M330](#), [M516](#)
- \@remove@tlig [I978](#), [I986](#)
- \@remove@tlig@ [I978](#), [I979](#)
- \@remove@tlig@@ [I979](#), [I982](#)
- \@remove@element [f32](#), [N272](#)
- \@remove@fromreset
..... [m46](#), [m69](#), [m72](#), [m91](#), [m92](#)
- \@reqcolroom [M1090](#), [M1091](#),
[M1094](#), [M1096](#), [M1097](#), [M1102](#),
[M1106](#), [M1108](#), [M1136](#), [M1137](#),
[M1243](#), [M1244](#), [M1248](#), [M1251](#),
[M1252](#), [M1257](#), [M1264](#), [M1266](#),
[M1298](#), [M1299](#), [M1410](#), [M1412](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- [M1414](#), [M1417](#), [M1419](#), [M1493](#),
[M1496](#), [M1499](#), [M1504](#), [M1506](#),
[M1986](#), [M2103](#), [M2108](#), [M2111](#)
\@resereda [t37](#)
\@reserveda [t36](#), [t135](#), [t136](#)
\@reservedb [t140](#), [t145](#), [t148](#)
\@reservedc [t142](#), [t147](#), [t149](#)
\@reset@options [N444](#), [N486](#), [N491](#)
\@resetactivechars [M575](#), [M598](#), [M656](#)
\@resetfps ... [M1205](#), [M1374](#), [M2053](#)
\@restorepar
... [69](#), [h5](#), [i287](#), [i303](#), [C127](#), [C135](#)
\@reversemarginfalse [M101](#), [I364](#)
\@reversemargintrue [I363](#)
\@rightmark [L16](#), [L51](#)
\@rightskip
... [A183](#), [A201](#), [C75](#), [D250](#), [D271](#)
\@rjfieldfalse [E34](#), [E65](#)
\@rjfieldtrue [E113](#)
\@rmfamilyhook [t104](#), [t116](#), [t205](#)
\@roman [m103](#), [m109](#)
\@rsbox [D369](#), [D376](#), [D380](#)
\@rtab [E59](#), [E74](#)
\@rule [D341](#), [D346](#), [D350](#)
\@sanitize ... [d386](#), [J7](#), [J18](#), [J24](#), [J35](#)
\@savebox [D77](#), [D84](#), [D90](#)
\@savemarbox [I326](#), [I327](#), [I330](#), [I333](#)
\@savepicbox [D77](#), [D84](#), [D94](#)
\@savsf [i97](#), [i103](#),
[i112](#), [i131](#), [i145](#), [i157](#), [i171](#), [i185](#)
\@savsk [i97](#), [i102](#),
[i113](#), [i132](#), [i146](#), [i158](#), [i172](#), [i186](#)
\@scolelt [M788](#), [M853](#)
\@sdblcplelt [M805](#), [M825](#), [M854](#)
\@secntformat [H60](#), [H111](#)
\@secondoftwo
... [a88](#), [d186](#), [d331](#), [d338](#), [d358](#),
[k257](#), [k296](#), [i131](#), [m156](#), [m161](#),
[x92](#), [x751](#), [x802](#), [x818](#), [z21](#), [L17](#),
[N46](#), [N66](#), [N78](#), [N115](#), [N133](#), [N1017](#)
\@secpentalty [i17](#), [H36](#), [H50](#)
\@sect [H54](#), [H55](#)
\@seqncr [B362](#)
\@setckpt [k209](#), [k216](#), [A16](#)
\@setfloattypecounts
... [M1074](#), [M1224](#), [M1395](#),
[M1476](#), [M1571](#), [M1692](#), [M2000](#)
\@setfontsize [t295](#)
\@setfps [I34](#)
\@setfpsbit [M2044](#),
[I73](#), [I75](#), [I77](#), [I85](#), [I143](#), [I146](#), [I149](#)
\@setmarks ... [M2214](#), [M2216](#), [M2231](#)
\@setminipage [D307](#), [I21](#), [I177](#), [I185](#), [I352](#)
\@setnobreak [I179](#), [I351](#)
\@setpar [69](#), [h3](#), [C78](#)
\@setref [z10](#)
\@setsize [t295](#)
\@settab [E59](#), [E81](#)
\@settodim [n17](#)
\@settopoint [n22](#)
\@setupverbvisiblespace
... [A274](#), [A292](#), [A295](#), [A307](#)
\@sffamilyhook [t109](#), [t116](#), [t206](#)
\@sharp .. [E184](#), [E211](#), [E241](#), [E256](#),
[E257](#), [E277](#), [E279](#), [E281](#), [E309](#)
\@shipoutsetup [M590](#)
\@shortstack [F43](#), [F44](#)
\@sline [F61](#), [F64](#), [F127](#)
\@slowromancap [m110](#), [m111](#)
\@spaces [g169](#)
\@specialoutput [M256](#)
\@specialpagefalse . [M97](#), [M607](#), [M665](#)
\@specialpagetrue [L9](#)
\@specialstyle [L9](#), [M607](#), [M665](#)
\@sptoken [d371](#), [d381](#)
\@sqrt [B309](#)
\@ssect [H53](#), [H112](#)
\@stackcr [F50](#), [F53](#)
\@star@or@long [d47](#), [d52](#),
[d99](#), [d121](#), [d127](#), [d153](#), [d162](#), [d195](#)
\@startcolumn [M263](#), [M270](#), [M775](#)
\@startdblcolumn [M775](#),
[M2236](#), [M2239](#), [M2276](#), [M2282](#)
\@startfield
... [E28](#), [E46](#), [E80](#), [E92](#), [E113](#), [E121](#)
\@startline . [E20](#), [E57](#), [E58](#), [E71](#), [E142](#)
\@startpbox [E181](#),
[E212](#), [E242](#), [E308](#), [E360](#), [E362](#)
\@startsection [H39](#)
\@starttoc [H149](#)
\@stopfield [E32](#), [E48](#), [E74](#),
[E81](#), [E113](#), [E115](#), [E128](#), [E130](#), [E142](#)
\@stopline [E30](#), [E56](#), [E73](#)
\@stpelt [m20](#), [m23](#)
\@strip@args [I76](#)
\@svector [F119](#), [F127](#)
\@sverb [A284](#), [A310](#), [A338](#), [A347](#), [A350](#)
\@svsec [H57](#), [H60](#), [H66](#), [H78](#)
\@svsechd [H76](#), [H101](#), [H121](#)
\@swaptwoargs [k318](#), [k320](#), [k329](#)
\@sxverbatim [A209](#), [A296](#), [A303](#)
\@tabacckludge .. [I223](#), [I225](#), [I451](#), [I452](#)
\@tabacol [E166](#), [E234](#)
\@tabarray [E158](#), [E168](#), [E169](#)
\@tabclassiv [E168](#), [E306](#)
\@tabclassz [E167](#), [E258](#)
\@tabcr [E56](#), [E61](#)
\@tabfbox [E16](#), [E68](#), [E70](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \@tablab E60, [E114](#)
- \@tabminus E60, [E105](#)
- \@tabplus E60, [E98](#)
- \@tabpush
 - [E11](#), [E65](#), [E73](#), [E128](#), [E131](#), [E133](#)
- \@tabrj E60, [E112](#)
- \@tabular [E162](#), [E165](#), [E166](#)
- \@tabularcrr [E168](#), [E196](#)
- \@tempa ... [z45](#), [z46](#), [N578](#), [N579](#), [N580](#)
- \@tempboxa [e13](#), [l69](#), [n17](#), [n18](#),
 - [C205](#), [C211](#), [C212](#), [C214](#), [D29](#),
[D30](#), [D31](#), [D32](#), [D37](#), [D38](#), [D39](#),
[D40](#), [D130](#), [D160](#), [D167](#), [D177](#),
[D298](#), [D321](#), [D386](#), [D387](#), [D388](#),
[D395](#), [D396](#), [D397](#), [D398](#), [F162](#),
[F163](#), [F233](#), [F234](#), [F257](#), [F262](#),
[F267](#), [F268](#), [F301](#), [F304](#), [F311](#),
[F312](#), [F333](#), [F334](#), [F339](#), [F340](#),
[F400](#), [F418](#), [H138](#), [H139](#), [M305](#),
[M377](#), [M382](#), [M383](#), [M424](#),
[M429](#), [M430](#), [M566](#), [M626](#),
[M633](#), [M634](#), [M685](#), [M692](#),
[M693](#), [M721](#), [M725](#), [M737](#),
[M743](#), [M750](#), [M751](#), [M752](#),
[M753](#), [M757](#), [M765](#), [I322](#), [I356](#)
- \@tempcnta [e7](#), [s739](#), [s740](#),
[s741](#), [s742](#), [s746](#), [E218](#), [E219](#),
[E220](#), [E221](#), [F67](#), [F68](#), [F94](#),
[F95](#), [F96](#), [F109](#), [F110](#), [F111](#),
[F112](#), [F114](#), [F115](#), [F128](#), [F129](#),
[F134](#), [F136](#), [F137](#), [F138](#), [F139](#),
[F140](#), [F143](#), [F145](#), [F146](#), [F147](#),
[F148](#), [F149](#), [F150](#), [F151](#), [F152](#),
[F153](#), [F154](#), [F184](#), [F185](#), [F186](#),
[F187](#), [F188](#), [F206](#), [F207](#), [F208](#),
[F209](#), [F210](#), [F211](#), [F224](#), [F225](#),
[F226](#), [F228](#), [F230](#), [F232](#), [F234](#),
[F271](#), [F276](#), [F315](#), [F319](#), [F335](#),
[F336](#), [F337](#), [F338](#), [F344](#), [F345](#),
[F346](#), [F347](#), [F348](#), [F349](#), [F391](#),
[F407](#), [M16](#), [M18](#), [M20](#), [M934](#),
[M935](#), [M936](#), [M937](#), [M957](#),
[M958](#), [M959](#), [M960](#), [M982](#),
[M985](#), [M1018](#), [M1021](#), [M1132](#),
[M1294](#), [M1574](#), [M1577](#), [M1695](#),
[M1698](#), [M1813](#), [M1815](#), [M1818](#),
[M1820](#), [M1822](#), [M1844](#), [M2034](#),
[M2035](#), [M2039](#), [M2045](#), [M2049](#),
[N596](#), [N598](#), [N599](#), [N600](#), [N711](#),
[N713](#), [N714](#), [N715](#), [Q160](#), [Q165](#),
[Q166](#), [Q167](#), [Q260](#), [Q262](#), [Q263](#),
[Q264](#), [Q271](#), [Q273](#), [Q274](#), [Q275](#),
[Q281](#), [Q283](#), [Q284](#), [Q285](#), [Q292](#),
[Q294](#), [Q295](#), [Q296](#), [Q322](#), [Q324](#),
[Q325](#), [Q326](#), [Q348](#), [Q350](#), [Q351](#),
[Q352](#), [Q358](#), [Q360](#), [Q361](#), [Q362](#),
[Q391](#), [Q396](#), [Q397](#), [Q398](#), [I62](#),
[I68](#), [I70](#), [I79](#), [I80](#), [I90](#), [I91](#), [I131](#),
[I137](#), [I139](#), [I152](#), [I153](#), [I159](#), [I160](#)
- \@tempcntb ... [e7](#), [s740](#), [s744](#), [s746](#),
[F137](#), [F138](#), [F139](#), [F141](#), [F142](#),
[F143](#), [F271](#), [F272](#), [F276](#), [F277](#),
[F315](#), [F316](#), [F319](#), [F320](#), [M17](#),
[M20](#), [M21](#), [M2045](#), [M2046](#),
[M2047](#), [Q161](#), [Q165](#), [Q392](#),
[Q396](#), [I88](#), [I89](#), [I90](#), [I157](#), [I158](#), [I159](#)
- \@tempdima . [e10](#), [o183](#), [o188](#), [p183](#),
[p188](#), [B173](#), [B176](#), [B182](#), [D43](#),
[D44](#), [D159](#), [D160](#), [D165](#), [D166](#),
[D167](#), [D169](#), [D220](#), [D221](#), [D296](#),
[D300](#), [D353](#), [D356](#), [D357](#), [D384](#),
[D386](#), [D392](#), [D395](#), [E35](#), [E36](#),
[E37](#), [E76](#), [E77](#), [E78](#), [E79](#), [E206](#),
[E207](#), [F90](#), [F91](#), [F93](#), [F94](#), [F95](#),
[F96](#), [F97](#), [F98](#), [F223](#), [F224](#),
[F225](#), [F234](#), [F258](#), [F259](#), [F263](#),
[F264](#), [F302](#), [F303](#), [F305](#), [F307](#),
[F336](#), [F338](#), [F343](#), [F344](#), [F345](#),
[H190](#), [H191](#), [H213](#), [H214](#), [H227](#),
[M229](#), [M230](#), [M231](#), [M487](#),
[M489](#), [M535](#), [M537](#), [M538](#),
[M543](#), [M548](#), [M552](#), [M557](#),
[M561](#), [M917](#), [M920](#), [M940](#),
[M950](#), [M962](#), [M972](#), [M1637](#),
[M1638](#), [M1641](#), [M1642](#), [M1762](#),
[M1763](#), [M1767](#), [M1768](#), [M1823](#),
[M1824](#), [M1825](#), [M1826](#), [M1829](#),
[M1832](#), [M1835](#), [M1837](#), [M2152](#),
[M2153](#), [M2155](#), [M2156](#), [I196](#),
[I198](#), [I218](#), [I220](#), [I258](#), [I259](#), [I260](#)
- \@tempdimb [e10](#), [o184](#),
[o189](#), [o559](#), [o563](#), [p184](#), [p189](#),
[p559](#), [p563](#), [q133](#), [q134](#), [q391](#),
[q414](#), [q415](#), [q424](#), [q425](#), [q429](#),
[q447](#), [q450](#), [q453](#), [q455](#), [D223](#),
[D224](#), [D354](#), [D357](#), [D385](#), [D387](#),
[D393](#), [D396](#), [F91](#), [F92](#), [F253](#),
[F255](#), [F256](#), [F298](#), [F299](#), [F300](#),
[F331](#), [F332](#), [F341](#), [F342](#), [M940](#),
[M941](#), [M942](#), [M943](#), [M950](#),
[M962](#), [M963](#), [M964](#), [M965](#), [M972](#)
- \@tempdimc
 - [e10](#), [q408](#), [q409](#), [q411](#), [q412](#),
[q414](#), [q415](#), [D355](#), [D356](#), [D357](#)
- \@tempskipa [e14](#),
[i44](#), [i47](#), [i48](#), [i235](#), [i242](#), [i244](#),
[i247](#), [q135](#), [q136](#), [C116](#), [C117](#),
[C118](#), [C150](#), [C152](#), [C153](#), [C154](#),

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=ltlfgs.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- C222, C223, C224, H42, H44,
H45, H50, H62, H63, H88, H89,
H91, H103, H104, H113, H114,
M1872, M1873, M1875, M1883
- \@tempskipb ... [e14](#), i194, i196, i198,
i201, i203, i213, i233, i235, i236,
i240, i242, i244, i245, i268, i271
- \@tempswafalse ... a78, b251, k185,
o58, o608, p58, p608, s281,
s336, s400, s481, s1065, s1071,
A18, A224, A245, K13, M988,
M1024, M1580, M1701, N574, N589
- \@tempswatrue ... a79, b257,
k183, k188, o61, o609, o610,
o613, o616, p61, p609, p610,
p613, p616, s284, s339, s403,
s484, s1028, A42, A229, A250,
K13, M1582, M1605, M1703,
M1728, M2113, M2130, N571
- \@temptokena ... [e16](#), A54, A58, A67, A80,
A81, L26, L27, L34, L35, L48, L49
- \@testdef ... A17, [A40](#)
- \@testfalse ... M12, M14, M15
- \@testfp ... M882, M902, M938,
M961, [M2037](#), M2166, M2183
- \@testopt ... d18, d54,
[d74](#), d78, d123, i7, i8, i9, i10, B349
- \@testpach ... E246, E324
- \@testpatch ... [E324](#)
- \@testtrue ... M13, M21, M356,
M885, M904, M944, M966, M2041
- \@testwrongwidth ... [M345](#), M883,
M939, M1112, M1426, M1631
- \@text@composite ... [l76](#), l1020, l1025
- \@text@composite@x ... [l76](#)
- \@textbottom ... L54,
L56, M511, M549, M563, [M572](#)
- \@textfloatsheight ...
M476, M1087, M1089, M1139,
M1140, M1145, M1240, M1242,
M1302, M1304, M1310, [M1986](#)
- \@textmin ... M112, M1089, M1093,
M1096, M1097, M1242, M1247,
M1251, M1252, M1414, M1499,
M1598, M1600, M1616, M1720,
M1722, M1740, M2094, M2096,
M2098, I285, I286, I299, I300
- \@textsubscript ... I387, [I388](#), I395
- \@textsuperscript ... I377, I379, [I380](#)
- \@texttop [L54](#), [L56](#), M507, M530, [M572](#)
- \@tf@r ... f25, f26
- \@tfor ... [f25](#), k258, k297, k391, w88,
D52, E244, F250, F294, I63, I132
- \@tforloop ... f27, f28, f30
- \@thanks ... H11, [H34](#)
- \@thefnmark . D331, I376, I377, I400,
I405, I415, I424, I429, I440, I445
- \@thefoot ... M124, M610,
M613, M640, M669, M672, M699
- \@thehead ... M123, M610,
M612, M630, M669, M671, M689
- \@themargin ... M74, M611,
M613, M625, M670, M672, M684
- \@themark L25, L26, L33, L34, L49, [L52](#)
- \@thirdofthree ... [d190](#), l197
- \@thm ... G12, G18, G24, [G26](#)
- \@thmcounter ... G11, G17, [G33](#)
- \@thmcountersep ... G10, [G33](#)
- \@title ... H7, [H31](#)
- \@tocrmarg ... H186, H209
- \@toodeep ... [g203](#), C36, C232, C243
- \@toplist ... M64,
M384, M385, M431, M432,
M716, M722, M732, M733,
M1025, M1037, M1921, M1948
- \@topnewpage ... [M199](#)
- \@topnum ... M105, M1022,
M1023, M1037, M1041, M1049,
M1458, M1463, M1551, M1558,
M1912, M1939, M1980, I271
- \@toproom ... M106, M1025,
M1037, M1913, M1940, I273
- \@topsep ... [C1](#), C71, C73, C171
- \@topsepadd .. [C1](#), C59, C61, C71, C124
- \@totalleftmargin ...
A220, A242, [C9](#), C53,
C54, D249, D270, E35, E64, E69
- \@trivlist ... C48, [C57](#), C92
- \@tryfcolumn ... M778,
M798, M816, [M832](#), M2167, M2184
- \@trylist ...
M841, M844, M877, M897, M919
- \@ttfamilyhook ... t114, [t116](#), t207
- \@twoclasseserror ... N301, [N554](#)
- \@twocolumnfalse ... M99, M147
- \@twocolumntrue ... M206
- \@twoloadclasserror ... N484, [N549](#)
- \@twosidefalse ... M100
- \@typein ... d17, d18, d25, d33
- \@typeset@protect ...
d77, d217, [d224](#), d226, l26, l32,
l210, l218, t296, A103, A121, Q254
- \@uclclist .. l1442, l1443, l1490, [Q461](#)
- \@undefined ...
a68, a69, a108, a109, a110,
a131, a139, a147, a154, a205,
a209, a235, a242, a326, a327,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

b65, b81, b105, b106, b121,
 b122, b127, b136, b149, b184,
 b189, b222, b223, b235, b245,
 b280, b448, b491, b537, b538,
 c49, c57, d19, d197, d301, d315,
 d333, d340, g28, i372, k58, k59,
 k113, k114, k245, k273, k274,
 k275, k276, k278, k283, k329,
 l195, l197, l333, l335, l337, l339,
 l341, l343, l345, l347, l349, l351,
 l370, l372, l374, l455, l694, l697,
 m170, o369, o409, o471, o504,
 o568, o575, p369, p409, p471,
 p504, p568, p575, q506, q507,
 r4, r5, r6, r7, r8, r9, r10,
 r11, r12, r13, r14, r15, r16,
 r17, r18, r19, r20, s597, s704,
 s804, t52, t176, t177, t178, t179,
 t180, t181, t182, t183, t184,
 t185, t186, t187, t205, t206,
 t207, t208, t209, t210, t253,
 t254, t255, t256, t257, t258,
 t269, u15, u48, u63, w37, w38,
 w39, w122, x52, x183, x298,
 x540, x541, x571, x572, x573,
 x575, x576, x577, x578, x579,
 x580, x583, x584, x585, x586,
 x589, x590, x591, x592, x594,
 x595, x596, x598, x599, x604,
 x605, x606, x607, x608, x609,
 x610, x611, x612, x613, x614,
 x615, x616, x617, x618, x619,
 x621, x622, x624, x625, x626,
 x627, x628, x629, x630, x631,
 x632, x633, x635, x636, x637,
 x638, x639, x640, x641, x642,
 x643, x645, x646, x647, x648,
 x649, x650, x651, x652, x653,
 x654, x655, x656, x657, x658,
 x659, x660, x661, x662, x663,
 x664, x665, x666, x667, x668,
 x669, x670, x671, x676, x677,
 x679, x680, x681, x682, x683,
 x684, x685, x686, x688, x689,
 x691, x692, x694, x695, x696,
 x697, x699, x700, x701, x703,
 x705, x706, x707, x708, x709,
 x710, x711, x713, x714, x715,
 x716, x717, x718, x719, x720,
 x721, z56, z57, A74, A75, A76,
 A77, A268, A305, A306, A307,
 A308, B237, B240, B283, B296,
 D21, D85, D149, D205, D347,
 D377, F290, F291, H224, K33,
 M36, M368, M369, N4, N82,
 N177, N474, N500, N609, N688,
 N691, N702, N703, N704, N705,
 N706, N707, N708, N779, N782,
 N798, N818, N825, P2, P13,
 P14, P15, P27, P28, P74, P84,
 P173, P181, P189, P197, P224,
 P225, P226, P227, P228, P229,
 P230, P231, P232, P233, P234,
 P235, P236, P237, P238, P239,
 P240, P241, P242, P248, Q10,
 Q18, Q25, Q40, Q59, Q68, Q75,
 Q93, Q94, Q201, Q214, Q226,
 Q247, Q248, Q249, Q278, Q280,
 Q319, Q320, Q339, Q340, Q341,
 Q342, Q343, Q344, Q345, Q346,
 Q347, Q364, Q380, Q381, Q382,
 Q430, Q431, Q491, Q527, Q528,
 Q529, Q530, Q531, I5, I394, I395
 \@unexpandable@protect
 d194, d229, d235, d240, k139, E240
 \@unknownoptionerror N495, N524, N537
 \@unprocessedoptions
 .. N267, N329, N471, N475, N539
 \@unused .. d4, g15, g32, g59, k3, N803
 \@unusedoptionlist k16, k18,
 k73, k75, N11, N220, N221, N273
 \@upline F155, F156, F162
 \@upordown . F75, F76, F84, F105, F131
 \@upvector F126, F162
 \@use@option
 .. N238, N250, N260, N262, N271
 \@use@text@encoding . I150, x16, x1054
 \@vbsphack i193
 \@verb A338, A347, A350
 \@verbatim .. A214, A260, A294, A303
 \@verbvisiblespacebox
 A279, A280, A283, A308
 \@vereq u449, u450
 \@viipt o651, p651
 \@vipt o650, p650
 \@vpt o649, p649
 \@vline F60, F155
 \@vobeyspaces
 .. A207, A260, A292, A296, A350
 \@vpt o648, p648
 \@vspace i280
 \@vspacer i280
 \@vtryfc M847, M855
 \@vvector F118, F126
 \@warning g166
 \@wckptelt k210, k213
 \@whiledim f7, F36, F83

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\@whilenoop` [f3](#)
`\@whilenum` [f3](#), [E220](#),
[F31](#), [F185](#), [F187](#), [F207](#), [F210](#), [F407](#)
`\@whilesw` [f10](#), [M264](#), [M394](#),
[M403](#), [M441](#), [M451](#), [M2237](#), [M2277](#)
`\@whileswnoop` [f10](#)
`\@wholewidth` [D115](#), [D117](#),
[D118](#), [D120](#), [D122](#), [D123](#), [D124](#),
[D125](#), [F2](#), [F38](#), [F40](#), [F42](#), [F157](#),
[F160](#), [F198](#), [F205](#), [F274](#), [F281](#),
[F318](#), [F324](#), [F365](#), [F366](#), [F404](#)
`\@width` [b394](#),
[d11](#), [i352](#), [l289](#), [l294](#), [q146](#),
[u610](#), [D120](#), [D122](#), [D173](#), [D180](#),
[D357](#), [D401](#), [E176](#), [E207](#), [E323](#),
[E342](#), [F107](#), [F157](#), [F160](#), [F177](#),
[F184](#), [F198](#), [F205](#), [F274](#), [F318](#),
[F404](#), [M1851](#), [M2226](#), [M2260](#), [I371](#)
`\@wrglossary` [J25](#), [J30](#)
`\@wrindex` [J8](#), [J13](#)
`\@writeckpt` [k198](#), [k207](#)
`\@writefile` [k30](#), [k88](#), [A61](#), [H177](#)
`\@writesetup` [M590](#)
`\@wrong@font@char` [l161](#),
[o472](#), [o506](#), [o519](#), [p472](#), [p506](#), [p519](#)
`\@wtryfc` [M857](#), [M867](#)
`\@x@protect` [d80](#), [d216](#)
`\@x@sf` [I434](#), [I436](#)
`\@xDeclareMathDelimiter` .. [s769](#), [s890](#)
`\@xaddvskip` [i193](#), [i214](#)
`\@xarg` [F57](#), [F60](#), [F65](#),
[F69](#), [F70](#), [F106](#), [F108](#), [F113](#),
[F114](#), [F118](#), [F124](#), [F132](#), [F350](#)
`\@xargarraycr` [E193](#), [E202](#), [E206](#)
`\@xargdef` [d55](#)
`\@xarraycr` [E190](#), [E191](#)
`\@xbitor` [M15](#), [M17](#)
`\@xcentercr` [A158](#), [A165](#), [A169](#)
`\@xdblarg` [d384](#)
`\@xdblfloat` [I264](#)
`\@xdim` [F26](#), [F32](#), [F34](#), [F354](#),
[F408](#), [F409](#), [F410](#), [F411](#), [F417](#)
`\@xeqncr` [B341](#)
`\@xexnoop` [E214](#), [E224](#)
`\@xexpast` [E215](#), [E216](#)
`\@xfloat` [I28](#), [I29](#), [I34](#), [I266](#)
`\@xfootnote` [I399](#), [I402](#)
`\@xfootnotemark` [I422](#), [I426](#)
`\@xfootnotenext` [I439](#), [I442](#)
`\@xhline` [E336](#), [E337](#)
`\@xifnch` [d372](#), [d382](#)
`\@xiipt` .. [o655](#), [p655](#), [u156](#), [u158](#), [u159](#)
`\@xipt` [o654](#), [p654](#), [u155](#)
`\@xivpt` [o656](#), [p656](#), [u157](#), [u159](#)
`\@xmpar` [I324](#), [I325](#)
`\@xnewline` [i60](#), [i61](#), [i72](#), [i73](#), [i80](#)
`\@xnext` [M10](#), [M11](#)
`\@xnthm` [G5](#), [G6](#)
`\@xobeysp` [i330](#), [A208](#), [A209](#), [A276](#), [A280](#)
`\@xprocess@ptions` . [N229](#), [N244](#), [N256](#)
`\@xpt` [o653](#), [p653](#), [u154](#), [u157](#), [u158](#)
`\@xssect` [H86](#), [H87](#), [H123](#)
`\@xtabcr` [E56](#), [E57](#)
`\@xtabularcr` [E197](#), [E198](#)
`\@xthm` [G28](#), [G29](#)
`\@xtryfc` [M844](#), [M872](#)
`\@xtypein` [d18](#), [d20](#), [d27](#)
`\@xverbatim` [A209](#), [A260](#)
`\@xvipt` [o657](#), [p657](#), [u158](#), [u160](#)
`\@xxDeclareMathDelimiter` .. [s754](#), [s758](#)
`\@xxpt` [o658](#), [p658](#), [u159](#), [u160](#)
`\@xxvpt` [o659](#), [p659](#), [u160](#)
`\@xxxii` [e2](#), [l422](#), [l424](#), [M879](#), [M880](#),
[M899](#), [M900](#), [M935](#), [M936](#),
[M958](#), [M959](#), [M2003](#), [I89](#), [I158](#)
`\@xympar` [I328](#), [I332](#), [I354](#)
`\@yarg` [F57](#), [F61](#), [F65](#), [F66](#), [F75](#), [F113](#),
[F119](#), [F126](#), [F128](#), [F155](#), [F350](#)
`\@yargarraycr` [E194](#), [E204](#), [E208](#)
`\@yargd@f` [d82](#)
`\@yargdef` [d59](#), [d69](#), [d82](#), [d98](#)
`\@ydim` [F27](#), [F32](#), [F34](#), [F355](#),
[F412](#), [F413](#), [F414](#), [F415](#), [F416](#)
`\@yeqncr` [B341](#)
`\@ympar` [I324](#), [I329](#)
`\@ynthm` [G5](#), [G14](#)
`\@ythm` [G28](#), [G29](#)
`\@ytryfc` [M890](#), [M909](#), [M913](#)
`\@yyarg` [F65](#), [F66](#), [F67](#), [F70](#), [F132](#), [F350](#)
`\@ztryfc` [M918](#), [M929](#)
`\[` [o347](#), [p347](#),
[t39](#), [t48](#), [B243](#), [B301](#), [B382](#), [Q422](#)
`\` [71](#), [a45](#), [a46](#),
[a74](#), [a247](#), [a248](#), [a249](#), [a250](#),
[a253](#), [a260](#), [a261](#), [a262](#), [a263](#),
[a266](#), [a273](#), [a274](#), [a275](#), [a276](#),
[a279](#), [a286](#), [a292](#), [a293](#), [a297](#),
[a299](#), [a300](#), [a304](#), [a309](#), [a310](#),
[a313](#), [a319](#), [b13](#), [d193](#), [d386](#),
[g227](#), [i52](#), [i378](#), [k390](#), [k405](#),
[l557](#), [o335](#), [p335](#), [u243](#), [A179](#),
[A183](#), [A187](#), [A211](#), [B323](#), [B442](#),
[D277](#), [D385](#), [D387](#), [E61](#), [E158](#),
[E168](#), [E182](#), [F50](#), [P259](#), [P411](#), [Q407](#)
`\{` [a3](#), [a7](#), [a74](#),
[b2](#), [b13](#), [g22](#), [l308](#), [l559](#), [o336](#),
[p336](#), [u241](#), [A210](#), [B59](#), [B154](#), [Q410](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- $\backslash}$. a8, a74, b3, b13, g21, l309, l560, o337, p337, u242, A210, B59, Q411
 $\backslash]$ b440, o348, p348, [B243](#), B302, [B406](#), [Q423](#)
 $\backslash^$ a63, a72, a75, a119, a330, b7, b9, b11, b14, b359, b360, b374, b375, d5, d387, i378, i380, i382, l231, l286, l382, l453, l463, l555, l638, l645, l649, l654, l659, l664, l671, l677, l678, l684, l689, l744, l1181, l1198, l1202, l1209, l1213, l1218, l1223, l1228, l1235, l1241, l1242, l1248, l1253, l1265, l1266, l1283, l1284, l1291, l1292, l1306, l1307, l1308, l1337, l1338, l1359, l1360, l1361, l1362, o333, o334, o339, p333, p334, p339, x188, x205, A48, A57, N603, N604, N605, N687, N690, N693, N718, N719, N720, N778, N781, N784, Q188, Q189, Q190, Q191, Q192, Q193, Q194, Q195, Q196, Q408, Q414, Q415, Q416, Q417, Q451, Q452, Q453, Q454, Q455, Q456, Q457, Q458, Q459
 $\backslash_$ a75, b8, b14, d387, l314, u246, B223, B224, Q409
 $\backslash^$ l232, l383, l417, l451, l461, l574, l636, l643, l647, l652, l657, l662, l669, l673, l674, l682, l687, l745, l787, l1179, l1200, l1207, l1211, l1216, l1221, l1226, l1233, l1237, l1238, l1246, l1251, o349, p349, t370, x191, x208, A351, D245, D266, E60, Q424
 $\backslash|$. l558, m135, m146, u565, u566, Q425
 $\backslash\sim$ a75, b10, b14, d387, g20, i332, l239, l287, l384, l464, l556, l639, l651, l655, l665, l681, l685, l746, l1182, l1199, l1203, l1215, l1219, l1229, l1245, l1249, l1293, l1294, l1295, l1347, l1348, x197, x215, A286, A312, A355, Q412
 $\backslash\sqcup$ a74, a91, b13, b359, b377, d386, g19, g20, g21, g22, g25, i331, o332, o582, o618, o643, p332, p582, p618, p643, u244, A207, A208, A291, G36, G38, K17, N186, Q402

A

 $\backslash A$ [Q185](#), [Q427](#), [Q448](#)
- $\backslash a$ [l223](#), [E1](#), Q176, Q428, Q439
 $\backslash AA$ [b365](#), l240, l425, l523
 $\backslash aa$ [b365](#), l245, l419, l533
 $\backslash abovedisplayshortskip$. . . b340, B450
 $\backslash abovedisplayskip$. . . b339, B443, B445, B447, B448, B449, B450
 $\backslash accent$ l73, l393, l420, l476, l757
 $\backslash accent@spacefactor$ l70, l73, l74
 $\backslash active$ a64, a119, a330, b10, b11, b374, b375, b377, A207, A208, A285, A291, A311, A353, B208, B223, M584, N603, N604, N605, N687, N690, N693, N718, N719, N720, N778, N781, N784
 $\backslash active@math@prime$. [B207](#), B208, M588
 $\backslash acute$ u508
 $\backslash add@accent$ l65, [l67](#)
 $\backslash add@percent@to@temptokena$ [A48](#), A64, A66, A75
 $\backslash add@unicode@accent$ l999, l1003
 $\backslash add_to_callback$ [591](#), [P648](#)
 $\backslash addcontentsline$ H70, H80, [H159](#), l16
 $\backslash addpenalty$ [i220](#), C124, C170, C175, H50, M338, M1153, M1319
 $\backslash addto@hook$ o116, o118, [o647](#), p116, p118, [p647](#), s263, s359, s363, s380, s504, s510, s518, s534, s537, s540, s1037, s1044, s1047
 $\backslash addtocontents$ H164, H171, [H174](#)
 $\backslash addtocounter$ [l45](#), [m6](#), m18
 $\backslash addtolength$ [l52](#), [n16](#), B445, B447
 $\backslash addtoversion$ r20, [r139](#)
 $\backslash addvspace$ [i207](#), A169, C124, C171, C172, C176, C224, H50
 $\backslash adjdemerits$ b319
 $\backslash AE$ l241, l398, l524, l762, l1078, l1379, Q480
 $\backslash ae$ l246, l401, l534, l766, l1084, l1380, Q480
 $\backslash afterassignment$ b390, b393, d230, d236, l212, l220, o263, p263, B186
 $\backslash aftergroup$ o55, o277, p55, p277, q156, q222, s114, s121, s129, w64, A289, A315, D103, M604, M605, M662, M663
 $\backslash aleph$ u300
 $\backslash alloc@$ b90, b91, b92, b93, b94, b95, b96, b97, b98, b99, [b226](#), o14, p14, P20, P24, P36
 $\backslash allocationnumber$ [b37](#), b57, b69, b71, b143, b144, b145, b195, b196, b228, b229, b230, b241, b242, b243, b260, b266, b272, b273, b286,

File Key: a=lttdirchk.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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- b287, b288, E4, E9, P50, P51,
 P52, P90, P204, Q44, Q45, Q46
 \allowbreak
 b397, d422, d423, d442, d444, B40
 \Alph 145, m106
 \alph 145, m105
 \alpha u260
 \alpha@elt
 s45, s267, s454, s556, s1036, s1037
 \alpha@list
 s41, s43, s276, s442, s454, s499,
 s554, s555, s1032, s1038, s1039
 \amalg u371
 \and 441, H14, H27
 \angle u329
 \approx u414
 \arabic 145, m102, G33
 \arccos B13
 \arcsin B10
 \arctan B16
 \arg B26
 \array E156
 \arraycolsep B326,
 B327, B455, B456, E234, E314
 \arrayrulewidth
 E300, E314, E322, E323,
 E335, E339, E342, E352, E354
 \arraystretch E174, E175, E318
 \Arrowvert u561
 \arrowvert u559
 \asciispace A267,
 A269, A272, A275, A276, A305
 \ast u224, u387
 \asympt u441
 \AtBeginDocument
 k54, k109, K34, K48, N509, Q544
 \AtBeginDvi M86
 \AtEndDocument A9, N509
 \AtEndOfClass B381, N509
 \AtEndOfPackage N267, N329, N509
 \atopwithdelims B57, B58, B59
 \attribute P78
 \attributedef P78, P210
 \attributezero P210
 \author 441, H8, H24, H32

B
 \b l233, l389, l472, l753, l1190
 \backslash u243, u582
 \bar u512
 \baselineskip
 b358, b388, b424, q140, q141,
 q142, q144, q145, u502, B158,
 B159, B178, B184, B188, D254,
 D273, E186, F47, F167, M242,
 M273, M622, M637, M681, M696
 \baselinestretch
 o254, p254, q118, q119, q138, q199
 \batchmode k369, k370, r106, Q500, Q521
 \begin g197, g199, k314, q7,
 u4, v4, A88, A89, B386, B398,
 H14, H17, N338, O3, Q385, Q389
 \belowdisplayskip .. b342, B449
 \belowdisplayskip b341, B448
 \best@size q392, q416, q422, q428
 \beta u261
 \bezier 414, F368,
 F369, F423, F424, F439, F441
 \bfdef@ult t73, t149
 \bfdefault
 t15, t73, t84, t191, u90, u101, u108
 \bfseries t13, t14, t77, t189,
 t190, w19, z13, G36, G38, K20
 \bfseries@rm . t60, t81, t155, t157, t177
 \bfseries@rm@kernel .. t63, t155, t180
 \bfseries@sf . t60, t82, t158, t160, t178
 \bfseries@sf@kernel .. t64, t158, t181
 \bfseries@tt . t60, t83, t161, t163, t179
 \bfseries@tt@kernel .. t65, t161, t182
 \bgroup b372
 \biblecite K7, K9, K10
 \bibdata K25, K29
 \bibitem K3
 \bibliography 474, K27
 \bibliographystyle 474, K32
 \bibstyle K25, K37
 \Big u615,
 u618, u627, u629, B44, B45, B46
 \big u616, u628, B41
 \bigbreak b404, d424, d445
 \bigcap u337
 \bigcirc u384
 \bigcup u338
 \Bigg u622, u631, B50, B51, B52
 \bigg u620, u630, B47, B48, B49
 \Biggl B50
 \biggl B47
 \Biggm B51
 \biggm B48
 \Biggr B52
 \biggr B49
 \Bigl B44
 \bigl B41
 \Bigm B45
 \bigm B42
 \bigodot u345
 \bigoplus u344
 \bigotimes u343

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\Bigr</code>	B46	<code>\bx@GG</code>	M41
<code>\bigr</code>	B43	<code>\bx@H</code>	M31, M58
<code>\bigskip</code>	b409, i310	<code>\bx@HH</code>	M41
<code>\bigskipamount</code>	b408, i312, i313, I367	<code>\bx@I</code>	M31, M58
<code>\bigskipcup</code>	u348	<code>\bx@II</code>	M41
<code>\bigtriangledown</code>	u353, u354	<code>\bx@J</code>	M31, M58
<code>\bigtriangleup</code>	u352, u355	<code>\bx@JJ</code>	M41
<code>\biguplus</code>	u336	<code>\bx@K</code>	M32, M59
<code>\bigvee</code>	u334	<code>\bx@KK</code>	M42
<code>\bigwedge</code>	u335	<code>\bx@L</code>	M32, M59
<code>\binoppenalty</code>	b310	<code>\bx@LL</code>	M42
<code>\bm@b</code>	D37	<code>\bx@M</code>	M32, M59
<code>\bm@c</code>	D37	<code>\bx@MM</code>	M42
<code>\bm@l</code>	D37	<code>\bx@N</code>	M32, M59
<code>\bm@r</code>	D37	<code>\bx@NN</code>	M42
<code>\bm@s</code>	D37	<code>\bx@O</code>	M33, M60
<code>\bm@t</code>	D37	<code>\bx@OO</code>	M43
<code>\bmod</code>	B35	<code>\bx@P</code>	M33, M60
<code>\boldmath</code>	j14, t288	<code>\bx@PP</code>	M43
<code>\bordermatrix</code>	B172	<code>\bx@Q</code>	M33, M60
<code>\bot</code>	u313	<code>\bx@QQ</code>	M43
<code>\botfigrule</code>	M742, M2325	<code>\bx@R</code>	M33, M60
<code>\botmark</code>	L50, M647, M706	<code>\bx@RR</code>	M43
<code>\bottomfraction</code>	M2294, I275	<code>\bx@S</code>	M38
<code>\bowtie</code>	u474	<code>\bx@SS</code>	M44
<code>\Box</code>	t324	<code>\bx@T</code>	M38
<code>\boxmaxdepth</code>	b333, F247, F292, F331, M486, M506, M546, M715, M724, M764	<code>\bx@TT</code>	M44
<code>\brace</code>	B59	<code>\bx@U</code>	M38
<code>\braced</code>	u545, u549, u550, u552, u554	<code>\bx@UU</code>	M44
<code>\bracelu</code>	u547, u551, u553	<code>\bx@V</code>	M38
<code>\bracerd</code>	u546, u551, u553	<code>\bx@VV</code>	M44
<code>\braceru</code>	u548, u550, u554	<code>\bx@W</code>	M39
<code>\bracevert</code>	u600	<code>\bx@WW</code>	M45
<code>\brack</code>	B58	<code>\bx@X</code>	M39
<code>\break</code>	b397, b402, d425, d446, i89	<code>\bx@XX</code>	M45
<code>\breve</code>	u513	<code>\bx@Y</code>	M39
<code>\brokenpenalty</code>	b315, o601, p601	<code>\bx@YY</code>	M45
<code>\buildrel</code>	u461, B149	<code>\bx@Z</code>	M39
<code>\bullet</code>	u373	<code>\bx@ZZ</code>	M25, M45, M55
<code>\bx@A</code>	M30, M57		
<code>\bx@AA</code>	M40		
<code>\bx@B</code>	M30, M57		
<code>\bx@BB</code>	M40		
<code>\bx@C</code>	M30, M57		
<code>\bx@CC</code>	M40		
<code>\bx@D</code>	M30, M57		
<code>\bx@DD</code>	M40		
<code>\bx@E</code>	M30, M57		
<code>\bx@EE</code>	M40		
<code>\bx@F</code>	M31, M58		
<code>\bx@FF</code>	M41		
<code>\bx@G</code>	M31, M58		
		C	
		<code>\c</code>	l234, l335, l337, l339, l341, l343, l345, l347, l349, l351, l372, l374, l392, l456, l475, l600, l602, l627, l629, l642, l668, l695, l698, l699, l700, l701, l702, l703, l704, l705, l706, l756, l1192, l1206, l1232, l1289, l1290, l1309, l1310, l1313, l1314, l1319, l1320, l1331, l1332, l1339, l1340, l1343, l1344, x187, x204
		<code>\c@bottomnumber</code>	M2292, I269, I274
		<code>\c@dbltopnumber</code>	M2299, I268, I283, I297
		<code>\c@enumi</code>	C227

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\c@enumii</code>	C227 , C227	<code>\cdot</code>	u386
<code>\c@enumiv</code>	C227	<code>\cdotp</code>	u494 , u500
<code>\c@equation</code>	B303 , B336 , B462	<code>\cdots</code>	u500
<code>\c@errorcontextlines</code>	g163	<code>\cdp@elt</code>	o60 , o80 , o91 , o92 , o113 , o116 , o118 , p60 , p80 , p91 , p92 , p113 , p116 , p118 , s201 , s283 , s338 , s402 , s483 , Q368 , Q369
<code>\c@footnote</code>	H12 , I373 , I428	<code>\cdp@list</code>	o62 , o78 , o92 , o120 , o121 , p62 , p78 , p92 , p120 , p121 , s219 , s285 , s340 , s404 , s485 , Q369
<code>\c@mpfootnote</code>	D303 , I375	<code>\center</code>	A172
<code>\c@ncel</code>	u453 , u454	<code>center</code> (environment)	A172
<code>\c@page</code>	y3 , y6 , y7 , M138 , M1818	<code>\centering</code> A172 , A177 , A178 , A193 , A195
<code>\c@secnumdepth</code> ..	H56 , H71 , H81 , H140	<code>\centerline</code>	D402
<code>\c@tocdepth</code>	H140 , H184 , H207	<code>\cf@encoding</code>	l34 , l41 , l44 , l51 , l154 , o220 , o230 , o240 , o261 , p220 , p230 , p240 , p261
<code>\c@topnumber</code>	M2288 , I267 , I271	<code>\ch@ck</code>	b206 , b207 , b208 , b209 , b227 , b237 , b238 , b239 , b240 , b268 , b270 , b282 , b283 , b284 , b285 , b291 , N614 , N635 , N728
<code>\c@totalnumber</code>	M2295 , I270 , I276	<code>\changes</code>	u113 , N560 , N562 , N563 , N564
<code>\cal</code>	t371	<code>\char</code>	d402 , l391 , l394 , l427 , l430 , l441 , l448 , l474 , l478 , l483 , l486 , l488 , l490 , l727 , l755 , l758 , l791 , l798 , l805 , l828 , l831 , l879 , l988 , l1001 , l1158 , t294 , x551 , x558 , x560 , x674 , A267 , A356 , B205 , F112 , F140 , F154 , F162 , F165 , F234 , F272 , F277 , F316 , F320 , F335 , F336 , F338 , F349
<code>\calculate@math@sizes</code> ..	o555 , p555 , q173	<code>\chardef</code>	a64 , a70 , a71 , b10 , b16 , b17 , b18 , b19 , b20 , b58 , b64 , b66 , b73 , b79 , b82 , b84 , b94 , b96 , b97 , b98 , b99 , b108 , b114 , b115 , b128 , b130 , b194 , b242 , b246 , b248 , b272 , b287 , b418 , b419 , b420 , e2 , k50 , l18 , o14 , p14 , E4 , E9 , N609 , N727 , P20 , P24 , P36 , P45 , P46 , P88 , P157 , P211 , Q28 , Q30 , Q34 , Q53 , Q104 , Q105 , Q106 , Q107 , Q108 , Q109 , Q110
<code>\call_callback</code>	592 , P626	<code>\chardef@text@cmd</code>	l3
<code>\callback.register</code>	P555	<code>\charsubdef</code>	Q249
<code>\callback_descriptions</code> ..	592 , P757	<code>\charzero</code>	P211
<code>\cap</code>	u364	<code>\check</code>	u514
<code>\capitalacute</code> l833 , x184 , x201 , x604 , x856	<code>\check@command</code>	d162 , d164
<code>\capitalbreve</code> l840 , x185 , x202 , x605 , x863	<code>\check@icl</code> w9 , w44 , w49 , w55 , w63 , w70 , w72
<code>\capitalcaron</code> l839 , x186 , x203 , x606 , x862	<code>\check@icr</code> w9 , w44 , w50 , w56 , w64 , w73 , w78
<code>\capitalcedilla</code> l826 , x187 , x204 , x607 , x853		
<code>\capitalcircumflex</code> l834 , x188 , x205 , x608 , x857		
<code>\capitaldieresis</code> l836 , x189 , x206 , x609 , x859		
<code>\capitaldotaccent</code> l842 , x190 , x207 , x610 , x865		
<code>\capitalgrave</code> l832 , x191 , x208 , x611 , x855		
<code>\capitalhungarumlaut</code> l837 , x192 , x209 , x612 , x860		
<code>\capitalmacron</code> l841 , x193 , x210 , x613 , x864		
<code>\capitalnewtie</code> l846 , x199 , x211 , x614 , x930 , x931		
<code>\capitalogonek</code> l829 , x194 , x212 , x615 , x854		
<code>\capitalring</code> l838 , x195 , x213 , x616 , x861		
<code>\capitaltie</code> l844 , x196 , x214 , x617 , x926 , x927		
<code>\capitaltilde</code> l835 , x197 , x215 , x618 , x858		
<code>\caption</code>	l4		
<code>\cases</code>	B153 , B154 , B164 , B166		
<code>\catcodetable</code>	P88 , P109		
<code>\catcodetable@atletter</code> ..	589 , P93 , P233		
<code>\catcodetable@initex</code> ..	589 , P93 , P230		
<code>\catcodetable@latex</code> ..	589 , P93 , P232		
<code>\catcodetable@string</code> ..	589 , P93 , P231		

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\check@mathfonts</code>	<code>\columnwidth</code>
. . . j5, l302, l328, l360, l1196,	. k22, k25, k26, k28, k79, k82,
o283, o285, p283, p285, q204, x588	k83, k86, D301, D328, M80,
<code>\check@nocorr@</code> w46	M144, M145, M146, M201,
<code>\check@range</code> q333, q334	M202, M203, M204, M205,
<code>\check@single</code> q332, q354	M1843, M1845, M2224, M2228,
<code>\CheckCommand</code> d162	M2256, M2262, I99, I168, I413
<code>\CheckEncodingSubset</code> . . . x16, x82,	<code>\conditionally@traceoff</code> g239
x130, x131, x132, x180, x182,	<code>\conditionally@traceon</code> g239
x290, x541, x745, x795, x851,	<code>\cong</code> u449
x852, x920, x1037, x1040, x1054	<code>\contentsline</code> H164, H171, H178
<code>\chi</code> u280	<code>\coprod</code> u333
<code>\choose</code> B57	<code>\copyright</code> l285, l316, t307
<code>\circ</code> u383	<code>\cos</code> B12
<code>\circle</code> F236, F329, F425, F442	<code>\cosh</code> B14
<code>\citation</code> K11, K19, K43	<code>\cot</code> B18
<code>\cite</code> 474, K12	<code>\coth</code> B19
<code>\cl@ckpt</code> k210, m35	<code>\count@</code> a66, a179, a180, a181, a186,
<code>\cl@page</code> y4	b41, b191, b192, b197, b199,
<code>\ClassError</code> g84	b205, b206, b207, b208, b209,
<code>\ClassInfo</code> g84	b210, b394, b395, c14, c15,
<code>\ClassWarning</code> g84	c16, c17, c18, c20, d144, d148,
<code>\ClassWarningNoLine</code> g84	o612, o618, o620, p612, p618,
<code>\cleaders</code> b438, u540, u543	p620, q22, q256, q258, q280,
<code>\cleardoublepage</code> M138	q281, s260, s262, s266, s585,
<code>\clearpage</code> k179,	s586, s587, s633, s634, s635,
k197, A12, A86, M125, M138,	s694, s695, s696, s742, s743,
M143, M200, M407, M410,	s744, s786, s787, s788, s794,
M414, M455, M461, M2171, M2188	s795, s796, s845, s846, s847,
<code>\cline</code> E343	s853, s854, s855, s901, s902,
<code>\clubpenalty</code> b312, k10, k23,	s903, s909, s910, s911, s969,
k80, o599, p599, C128, C194,	s970, s971, s977, s978, s979,
C196, H100, H106, H130, H135	w115, w118, F406, F407, F408,
<code>\clubsuit</code> u323	F411, F412, F415, F419, N659,
<code>\col@number</code> M95, M148, M208, M220	N661, N662, N663, N749, N751,
<code>\colon</code> u495	N752, N753, Q172, Q173, Q180,
<code>\color@begingroup</code>	Q182, Q435, Q436, Q443, Q445
. o578, o638, p578, p638,	<code>\countdef</code> a66, b37, b38, b39,
B104, B134, D29, D63, D131,	b41, b51, b90, y3, P75, P85,
D299, D332, E47, E51, M491, I417	P174, P182, P190, P198, P212, Q61
<code>\color@endbox</code> D63, M224, M631,	<code>\counterwithin</code> 145, m77, m97
M641, M690, M700, I253, I344	<code>\counterwithin@s</code> m77, m78, m98
<code>\color@endgroup</code>	<code>\counterwithin@x</code> m77, m80, m99
. o583, o644, p583, p644, B104,	<code>\counterwithout</code> 145, m67, m94
B134, D29, D63, D89, D110,	<code>\counterwithout@s</code> m67, m68, m95
D133, D319, D335, E49, M495, I420	<code>\counterwithout@x</code> m67, m70, m96
<code>\color@hbox</code>	<code>\CountZero</code> P212
. D63, M628, M638, M687, M697	<code>\cr</code> b368, l497,
<code>\color@setgroup</code> D63, D89, D108	l503, l513, l519, x122, x126,
<code>\color@vbox</code>	x843, x847, B175, B179, B331,
. D63, M215, I96, I165, I335, I357	B361, B459, E180, E191, E198,
<code>\columnsep</code> k25, k82, M81, M202	E207, E208, E353, F52, F54, F55
<code>\columnseprule</code> M82, M2226, M2260	<code>\crrc</code> b425, l327, l362, l363, l390, l394,
	l397, l473, l477, l481, l483, l486,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

1726, 1754, 1758, 1761, 1828, 1831, 1878, 11195, t309, u329, u330, u332, u451, u454, u458, u522, u523, u524, u525, u526, u527, u529, u530, u531, u532, u533, u535, x127, x848, B155, B157, B158, B159, B175, B177, B178, B179, B197, B198, E159, E160, F52	\cyrchrdsc	11446
	\CYRCHVCRS	11447
	\cyrchvcrs	11447
	\CYRD	11447
	\cyrd	11447
	\CYRDELTA	11447
	\cyrdelta	11447
	\CYRDJE	11448
\create_callback 592, P607	\cyrdje	11448
\cs u113	\CYRDZE	11448
\csc B21	\cyrdze	11448
\cup u365	\CYRDZHE	11448
\curr@fontshape l180, o52, o298, o306, o310, o312, o454, o460, o463, o472, o479, o481, o489, o495, o498, o506, o513, o515, p52, p298, p306, p310, p312, p454, p460, p463, p472, p479, p481, p489, p495, p498, p506, p513, p515, q92, q100, q121, q431, q451, q483, q499, q514, s223, s228, t231, t240	\cyrdzhe	11448
	\CYRE	11448
	\cyre	11448
	\CYREPS	11449
	\cyreps	11448
	\CYREREV	11449
	\cyrerev	11449
	\CYRERY	11449
	\cyrery	11449
	\CYRF	11449
\curr@math@size o287, p287, q210, q216, q221, q238	\cyrf	11449
	\CYRFITA	11450
\CurrentOption l1472, l1474, l1486, N13, N222, N232, N233, N234, N239, N246, N247, N248, N251, N258, N259, N263, N264, N265, N272, N274, N283, N284, N285, N286, N293, N294, N295, N296, N443, N526, N527, N536, N537	\cyrfita	11449
	\CYRG	11450
	\cyrghcrs	11450
	\CYRGDSC	11450
	\cyrghdsc	11450
	\CYRGDSCHCRS	11450
	\cyrghdschcrs	11450
\CYRA l1444	\CYRGHCRS	11451
\cyra l1444, l1489	\cyrghcrs	11451
\CYRABHCH l1444	\CYRGHK	11451
\cyrabhch l1444	\cyrghk	11451
\CYRABHCHDSC l1444	\CYRGUP	11451
\cyrabhchdsc l1444	\cyrgup	11451
\CYRABHDZE l1445	\CYRH	11451
\cyrabhdze l1444	\cyrh	11451
\CYRABHHA l1445	\CYRHDSC	11452
\cyrabhha l1445	\cyrhdsc	11452
\CYRAE l1445	\CYRHHCRS	11452
\cyrae l1445	\cyrhhcrs	11452
\CYRB l1445	\CYRHHK	11452
\cyrb l1445	\cyrhhk	11452
\CYRBYUS l1446	\CYRHRSN	11453
\cyrbyus l1445	\cyrhrdsn	11452
\CYRC l1446	\CYRI	11453
\cyrc l1446	\cyri	11453
\CYRCH l1446	\CYRIE	11453
\cyrch l1446	\cyrie	11453
\CYRCHLDSC l1446	\CYRII	11453
\cyrchldsc l1446	\cyrii	11453
\CYRCHRDSC l1447	\CYRISHRT	11453

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

\cyrishrt	11453	\cyrq	11460
\CYRISHRTDSC	11454	\CYRR	11460
\cyrishrtdsc	11454	\cyrr	11460
\CYRIZH	11454	\CYRRDSC	11460
\cyrizh	11454	\cyrrdsc	11460
\CYRJE	11454	\CYRRHK	11461
\cyrje	11454	\cyrrhk	11460
\CYRK	11454	\CYRRTICK	11461
\cyrk	11454	\cyrrtick	11461
\CYRKBEAK	11455	\CYRS	11461
\cyrkbeak	11455	\cyrs	11461
\CYRKDSC	11455	\CYRSACRS	11461
\cyrkdsc	11455	\cyrsacrs	11461
\CYRKHCRS	11455	\CYRSCHWA	11462
\cyrkhcrs	11455	\cyrschwa	11462
\CYRKHK	11456	\CYRSDSC	11462
\cyrkhk	11455	\cyrsdsc	11462
\CYRKVCRS	11456	\CYRSEMISFTSN	11462
\cyrkvcrs	11456	\cyrsemisftsn	11462
\CYRL	11456	\CYRSFTSN	11463
\cyr1	11456	\cyrsftsn	11463
\CYRLDSC	11456	\CYRSH	11463
\cyrldsc	11456	\cyrsh	11463
\CYRLHK	11457	\CYRSHCH	11463
\cyr1hk	11456	\cyrshch	11463
\CYRLJE	11457	\CYRSHHA	11463
\cyr1je	11457	\cyrshha	11463
\CYRM	11457	\CYRT	11464
\cyrm	11457	\cyrt	11464
\CYRMDSC	11457	\CYRTDSC	11464
\cyrmdsc	11457	\cyrtldsc	11464
\CYRMHK	11457	\CYRTETSE	11464
\cyrmhk	11457	\cyrtetse	11464
\CYRN	11458	\CYRTSHE	11464
\cyrn	11458	\cyrtshe	11464
\CYRNDSC	11458	\CYRU	11465
\cyrndsc	11458	\cyr	11465
\CYRNG	11458	\CYRUSHRT	11465
\cyrng	11458	\cyrushrt	11465
\CYRNHK	11458	\CYRV	11465
\cyrnhk	11458	\cyrv	11465
\CYRNJE	11459	\CYRW	11465
\cyrnje	11458	\cyrw	11465
\CYRNLHK	11459	\CYRY	11465
\cyrnlhk	11459	\cyry	11465
\CYRO	11459	\CYRYA	11466
\cyro	11459	\cyrya	11466
\CYROTLD	11459	\CYRYAT	11466
\cyrotld	11459	\cyryat	11466
\CYRP	11459	\CYRYHCRS	11466
\cyrp	11459	\cyryhcrs	11466
\CYRPHK	11460	\CYRYI	11466
\cyrphk	11460	\cyryi	11466
\CYRQ	11460	\CYRYO	11467

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\cyrro</code>	l1466	x352, x353, x354, x355, x356,
<code>\CYRYU</code>	l1467	x357, x358, x359, x360, x361,
<code>\cyrYu</code>	l1467	x362, x363, x364, x365, x366,
<code>\CYRZ</code>	l1467	x367, x368, x369, x370, x371,
<code>\cyrz</code>	l1467	x372, x373, x374, x375, x376,
<code>\CYRZDSC</code>	l1467	x377, x378, x379, x380, x381,
<code>\cyrzdsc</code>	l1467	x382, x383, x384, x385, x386,
<code>\CYRZH</code>	l1467	x387, x388, x389, x390, x391,
<code>\cyrzh</code>	l1467	x392, x393, x394, x395, x396,
<code>\CYRZHDSC</code>	l1468	x397, x398, x399, x400, x401,
<code>\cyrzhdsc</code>	l1468	x402, x403, x404, x405, x406,
		x407, x408, x409, x410, x411,
		x412, x413, x414, x415, x416,
		x417, x418, x419, x420, x421,
		x422, x423, x424, x425, x426,
		x427, x428, x429, x430, x431,
		x432, x433, x434, x435, x436,
		x437, x438, x439, x440, x441,
		x442, x443, x444, x445, x446,
		x447, x448, x449, x450, x451,
		x452, x453, x454, x455, x456,
		x457, x458, x459, x460, x461,
		x462, x463, x464, x465, x466,
		x467, x468, x469, x470, x471,
		x472, x473, x474, x475, x476,
		x477, x478, x479, x480, x481,
		x482, x483, x484, x485, x486,
		x487, x488, x489, x490, x491,
		x492, x493, x494, x495, x496,
		x497, x498, x499, x500, x501,
		x502, x503, x504, x505, x538,
		x540, x736, x737, x738, x739,
		x778, x785, x786, x787, x788,
		x1064, x1065, x1066, x1067,
		x1068, x1069, x1070, x1071,
		x1072, x1073, x1074, x1075,
		x1076, x1077, x1078, x1079,
		x1080, x1081, x1082, x1083,
		x1084, x1085, x1086, x1087,
		x1088, x1089, x1090, x1091,
		x1092, x1093, x1094, x1095,
		x1096, x1097, x1098, x1099,
		x1100, x1101, x1102, x1103,
		x1104, x1105, x1106, x1107,
		x1108, x1109, x1110, x1111,
		x1112, x1113, x1114, x1115,
		x1116, x1117, x1118, x1119,
		x1120, x1121, x1122, x1123, x1124
<code>\DeclareEncodingSubset@aux</code>		x62, x64, x80
<code>\DeclareErrorFont</code>		o412, p412, s232, t333, u47
<code>\DeclareFixedFont</code>		o39, p39
<code>\DeclareFontEncoding</code>		l377,
D		
<code>\d</code>	l235, l395, l479,	
	l759, l1191, l1399, l1400, l1401,	
	l1402, l1405, l1406, l1407, l1408,	
	l1409, l1410, l1411, l1412, l1413,	
	l1414, l1415, l1416, l1417, l1418,	
	l1419, l1420, l1421, l1422, l1423,	
	l1424, l1425, l1426, l1427, l1428,	
	l1429, l1430, l1431, l1432, l1433,	
	l1434, l1435, l1436, l1437, l1438	
<code>\dag</code>	l312	
<code>\dagger</code>	l312,	
	m131, m137, m145, m146, u367	
<code>\dashbox</code>	F167, F426, F443	
<code>\dashv</code>	u395	
<code>\date</code>	441, H9, H25	
<code>\day</code>	a185, c11, N655, N745	
<code>\dblfigrule</code>	M767, M2325	
<code>\dblfloatpagefraction</code>		
	M2302, I287, I301	
<code>\dblfloatsep</code>	M753,	
	M765, M1640, M1766, M2309	
<code>\dbltextfloatsep</code> ...	M222, M230,	
	M769, M1639, M1765, M2309	
<code>\dbltopfraction</code> ...	M2301, I284, I298	
<code>\ddag</code>	l313	
<code>\ddagger</code>	l313,	
	m132, m138, m145, m147, u366	
<code>\ddot</code>	u510	
<code>\ddots</code>	u505	
<code>\deadcycles</code>	k203, A39, A86, M299	
<code>\declare@robustcommand</code>	d195	
<code>\DeclareCurrentRelease</code>		
	x729, x732, N980	
<code>\DeclareEmphSequence</code>		
	261, t216, t252, t253, t265	
<code>\DeclareEncodingSubset</code>	x60,	
	x61, x328, x329, x330, x331,	
	x332, x333, x334, x335, x336,	
	x337, x338, x339, x340, x341,	
	x342, x343, x344, x345, x346,	
	x347, x348, x349, x350, x351,	

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

1460, 1710, 1732, 1738, 1824, 1998,
 o82, p82, u118, u119, u120, u121
 \DeclareFontEncoding@
 o86, o88, o103,
 p86, p88, p103, Q279, Q299, Q365
 \DeclareFontEncoding@saved
 Q279, Q299, Q381
 \DeclareFontEncodingDefaults
 o132, p132, r90, r91, u34
 \DeclareFontFamily . o57, p57, r85, r86
 \DeclareFontFamilySubstitution
 o372, p372
 \DeclareFontSeriesDefault
 253, t34, t35, t174, t176
 \DeclareFontShape o18, o20, o381,
 o382, o383, o384, o385, o386,
 o387, o388, o389, o390, o391,
 o392, o393, o394, o395, o396,
 o397, o398, o399, o400, o401,
 p18, p20, p381, p382, p383,
 p384, p385, p386, p387, p388,
 p389, p390, p391, p392, p393,
 p394, p395, p396, p397, p398,
 p399, p400, p401, r25, r27, r81, r82
 \DeclareFontShape@ . o21, o22, p21, p22
 \DeclareFontSubstitution . . . l739,
 l825, o105, p105, u24, u32,
 u35, u36, u122, u123, u124, u125
 \DeclareMathAccent s573, u508, u509,
 u510, u511, u512, u513, u514,
 u515, u516, u517, u518, u519, u520
 \DeclareMathAlphabet
 r119, r123, r125, r132,
 s399, s562, u143, u144, u145, u146
 \DeclareMathAlphabetCharacter . s748
 \DeclareMathDelimiter
 s750, u247, u248, u249,
 u250, u251, u252, u255, u257,
 u258, u555, u557, u559, u561,
 u563, u566, u568, u570, u572,
 u574, u576, u578, u580, u582,
 u584, u586, u588, u590, u592,
 u594, u596, u598, u600, u602, u604
 \DeclareMathRadical s949, u521
 \DeclareMathSizes . . . o169, o175,
 o197, p169, p175, p197, u149,
 u150, u151, u152, u153, u154,
 u155, u156, u157, u158, u159, u160
 \DeclareMathSizes* o169, p169
 \DeclareMathSymbol s686, s749,
 s766, u161, u162, u163, u164,
 u165, u166, u167, u168, u169,
 u170, u171, u172, u173, u174,
 u175, u176, u177, u178, u179,
 u180, u181, u182, u183, u184,
 u185, u186, u187, u188, u189,
 u190, u191, u192, u193, u194,
 u195, u196, u197, u198, u199,
 u200, u201, u202, u203, u204,
 u205, u206, u207, u208, u209,
 u210, u211, u212, u213, u214,
 u215, u216, u217, u218, u219,
 u220, u221, u222, u223, u224,
 u225, u226, u227, u228, u229,
 u230, u231, u232, u233, u234,
 u235, u236, u237, u238, u239,
 u240, u241, u242, u243, u253,
 u254, u256, u260, u261, u262,
 u263, u264, u265, u266, u267,
 u268, u269, u270, u271, u272,
 u273, u274, u275, u276, u277,
 u278, u279, u280, u281, u282,
 u283, u284, u285, u286, u287,
 u288, u289, u290, u291, u292,
 u293, u294, u295, u296, u297,
 u298, u299, u300, u301, u302,
 u303, u304, u305, u306, u307,
 u308, u309, u310, u311, u312,
 u313, u314, u315, u316, u317,
 u319, u320, u321, u322, u323,
 u324, u325, u326, u333, u334,
 u335, u336, u337, u338, u339,
 u341, u342, u343, u344, u345,
 u346, u348, u349, u350, u351,
 u352, u353, u356, u357, u358,
 u359, u362, u363, u364, u365,
 u366, u367, u368, u369, u370,
 u371, u372, u373, u374, u375,
 u376, u377, u378, u379, u380,
 u381, u382, u383, u384, u385,
 u386, u387, u388, u389, u390,
 u391, u392, u393, u394, u395,
 u396, u397, u398, u399, u400,
 u401, u402, u403, u406, u407,
 u410, u411, u412, u413, u414,
 u415, u416, u417, u418, u419,
 u420, u421, u422, u424, u425,
 u426, u427, u428, u429, u430,
 u433, u434, u435, u437, u438,
 u439, u440, u441, u442, u443,
 u444, u445, u446, u447, u469,
 u471, u493, u494, u495, u545,
 u546, u547, u548, u606, u607, u608
 \DeclareMathVersion s245, t2, t3
 \DeclareOldFontCommand . . w125, w141
 \DeclareOption . . . 556, l1471, q29,
 q37, q45, q53, q56, q60, x736,
 x737, x738, x739, x741, x742,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

x743, x744, x785, x786, x787,
 x788, x789, x791, x792, [N205](#), [N547](#)
 \DeclareOption* [556](#), [N205](#)
 \DeclarePreloadSizes
 [o149](#), [p149](#), [r95](#), [r96](#), [v19](#), [v21](#),
 [v22](#), [v23](#), [v25](#), [v26](#), [v27](#), [v28](#),
 [v29](#), [v30](#), [v34](#), [v38](#), [v43](#), [v45](#),
 [v49](#), [v50](#), [v53](#), [v54](#), [v57](#), [v58](#), [v64](#)
 \DeclareRelease ... [x728](#), [x731](#), [N887](#)
 \DeclareRobustCommand
 .. [d195](#), [d392](#), [d393](#), [d400](#), [g4](#),
 [g11](#), [g30](#), [g57](#), [i7](#), [i8](#), [i9](#), [i10](#), [i11](#),
 [i68](#), [i79](#), [i280](#), [i316](#), [i330](#), [i335](#),
 [i350](#), [i361](#), [i362](#), [i363](#), [j2](#), [j3](#),
 [j13](#), [k239](#), [k315](#), [l150](#), [l158](#), [l307](#),
 [l308](#), [l309](#), [l310](#), [l311](#), [l312](#), [l313](#),
 [l314](#), [l316](#), [l318](#), [l320](#), [m155](#),
 [n19](#), [n20](#), [n21](#), [o215](#), [o243](#), [o244](#),
 [o245](#), [o246](#), [o251](#), [o253](#), [o271](#),
 [o576](#), [o585](#), [p215](#), [p243](#), [p244](#),
 [p245](#), [p246](#), [p251](#), [p253](#), [p271](#),
 [p576](#), [p585](#), [q113](#), [t4](#), [t7](#), [t10](#),
 [t13](#), [t16](#), [t19](#), [t22](#), [t25](#), [t28](#), [t77](#),
 [t89](#), [t101](#), [t106](#), [t111](#), [t189](#), [t192](#),
 [t195](#), [t198](#), [t201](#), [t224](#), [t225](#),
 [t260](#), [t266](#), [t288](#), [t290](#), [t294](#),
 [t307](#), [t311](#), [u327](#), [u328](#), [u329](#),
 [u340](#), [u347](#), [u404](#), [u405](#), [u436](#),
 [u448](#), [u452](#), [u455](#), [u460](#), [u462](#),
 [u464](#), [u467](#), [u470](#), [u472](#), [u473](#),
 [u475](#), [u477](#), [u479](#), [u481](#), [u483](#),
 [u485](#), [u487](#), [u489](#), [u491](#), [u497](#),
 [u499](#), [u501](#), [u504](#), [u522](#), [u525](#),
 [u528](#), [u532](#), [u536](#), [u539](#), [u542](#),
 [u549](#), [u552](#), [u609](#), [u610](#), [u611](#),
 [u616](#), [u618](#), [u620](#), [u622](#), [w3](#),
 [w126](#), [x4](#), [x11](#), [x45](#), [x1049](#), [z45](#),
 [A93](#), [A178](#), [A182](#), [A186](#), [A267](#),
 [A271](#), [B3](#), [B4](#), [B5](#), [B6](#), [B7](#),
 [B8](#), [B9](#), [B10](#), [B11](#), [B12](#), [B13](#),
 [B14](#), [B15](#), [B16](#), [B17](#), [B18](#), [B19](#),
 [B20](#), [B21](#), [B22](#), [B23](#), [B24](#), [B25](#),
 [B26](#), [B27](#), [B28](#), [B29](#), [B30](#), [B31](#),
 [B32](#), [B33](#), [B34](#), [B35](#), [B39](#), [B41](#),
 [B42](#), [B43](#), [B44](#), [B45](#), [B46](#), [B47](#),
 [B48](#), [B49](#), [B50](#), [B51](#), [B52](#), [B81](#),
 [B82](#), [B83](#), [B84](#), [B126](#), [B154](#),
 [B156](#), [B160](#), [B205](#), [B228](#), [B230](#),
 [B246](#), [B257](#), [B307](#), [B308](#), [B309](#),
 [B366](#), [B383](#), [B407](#), [D7](#), [D24](#),
 [D75](#), [D88](#), [D111](#), [D112](#), [D128](#),
 [D139](#), [D193](#), [D341](#), [D359](#), [D367](#),
 [D403](#), [D404](#), [D405](#), [D406](#), [D407](#),
 [E127](#), [E130](#), [E142](#), [F37](#), [F39](#),
 [F41](#), [H7](#), [H8](#), [H9](#), [H10](#), [H14](#),
 [H201](#), [K12](#), [L22](#), [L30](#), [L53](#), [L55](#),
 [L57](#), [L64](#), [M87](#), [N514](#), [N515](#),
 [N1010](#), [Q461](#), [Q471](#), [I378](#), [I386](#)
 \DeclareSizeFunction
 [q371](#), [q444](#), [q445](#),
 [q456](#), [q457](#), [q461](#), [q462](#), [q468](#),
 [q469](#), [q497](#), [q511](#), [q512](#), [q519](#), [q520](#)
 \DeclareSymbolFont
 [r136](#), [s280](#), [u133](#), [u134](#), [u135](#), [u136](#)
 \DeclareSymbolFontAlphabet
 [s1023](#), [u140](#), [u141](#), [u142](#)
 \DeclareSymbolFontAlphabet@
 [s1024](#), [s1027](#)
 \DeclareTextAccent [l64](#), [l378](#),
 [l379](#), [l380](#), [l381](#), [l382](#), [l383](#), [l384](#),
 [l385](#), [l386](#), [l387](#), [l388](#), [l461](#), [l462](#),
 [l463](#), [l464](#), [l465](#), [l466](#), [l467](#), [l468](#),
 [l469](#), [l470](#), [l471](#), [l735](#), [l740](#), [l741](#),
 [l742](#), [l743](#), [l744](#), [l745](#), [l746](#), [l747](#),
 [l748](#), [l749](#), [l750](#), [l832](#), [l833](#), [l834](#),
 [l835](#), [l836](#), [l837](#), [l838](#), [l839](#), [l840](#),
 [l841](#), [l842](#), [l843](#), [l844](#), [l845](#), [l846](#)
 \DeclareTextAccentDefault
 [l185](#), [l226](#), [l227](#), [l228](#),
 [l229](#), [l230](#), [l231](#), [l232](#), [l233](#), [l234](#),
 [l235](#), [l236](#), [l237](#), [l238](#), [l239](#), [l279](#),
 [l282](#), [x601](#), [x602](#), [x853](#), [x854](#),
 [x855](#), [x856](#), [x857](#), [x858](#), [x859](#),
 [x860](#), [x861](#), [x862](#), [x863](#), [x864](#), [x865](#)
 \DeclareTextCommand
 [l3](#), [l58](#), [l65](#), [l83](#), [l389](#),
 [l392](#), [l395](#), [l411](#), [l412](#), [l419](#), [l421](#),
 [l423](#), [l425](#), [l431](#), [l433](#), [l435](#), [l442](#),
 [l472](#), [l475](#), [l479](#), [l482](#), [l484](#), [l487](#),
 [l489](#), [l491](#), [l507](#), [l724](#), [l751](#), [l753](#),
 [l756](#), [l759](#), [l792](#), [l799](#), [l826](#), [l829](#),
 [l876](#), [l1003](#), [l1029](#), [l1031](#), [l1033](#),
 [l1157](#), [l1194](#), [x307](#), [x308](#), [x309](#),
 [x310](#), [x311](#), [x312](#), [x313](#), [x314](#),
 [x315](#), [x316](#), [x545](#), [x552](#), [x559](#), [x673](#)
 \DeclareTextCommandDefault
 [l57](#), [l186](#),
 [l188](#), [l283](#), [l286](#), [l287](#), [l288](#), [l290](#),
 [l292](#), [l296](#), [l300](#), [l301](#), [l303](#), [l304](#),
 [l305](#), [l306](#), [l326](#), [l355](#), [x179](#),
 [x181](#), [x184](#), [x185](#), [x186](#), [x187](#),
 [x188](#), [x189](#), [x190](#), [x191](#), [x192](#),
 [x193](#), [x194](#), [x195](#), [x196](#), [x197](#),
 [x198](#), [x199](#), [x218](#), [x219](#), [x220](#),
 [x221](#), [x222](#), [x223](#), [x224](#), [x225](#),
 [x226](#), [x227](#), [x228](#), [x229](#), [x230](#),
 [x231](#), [x232](#), [x233](#), [x234](#), [x235](#),
 [x236](#), [x237](#), [x238](#), [x239](#), [x240](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx


```

x241, x242, x243, x244, x245,
x246, x247, x248, x249, x250,
x251, x252, x253, x254, x255,
x256, x257, x258, x259, x260,
x261, x262, x263, x264, x265,
x266, x268, x269, x270, x271,
x272, x273, x274, x275, x276,
x277, x278, x279, x280, x281,
x282, x283, x284, x285, x286,
x287, x288, x289, x291, x292,
x293, x294, x295, x296, x297,
x574, x581, x582, x587, x593,
x919, x921, x922, x924, x926,
x928, x930, x932, x934, x936,
x938, x940, x942, x944, x946,
x948, x950, x952, x954, x956,
x958, x960, x962, x964, x966,
x968, x970, x972, x974, x976,
x978, x980, x982, x984, x986,
x988, x990, x992, x994, x996,
x998, x1000, x1002, x1004,
x1006, x1008, x1010, x1012,
x1014, x1016, x1018, x1020,
x1022, x1024, x1026, x1028,
x1030, x1032, x1034, x1036, x1039

\DeclareTextComposite .....
..... 176, 1449, 1450, 1581,
1582, 1583, 1584, 1585, 1586, 1587,
1588, 1589, 1590, 1591, 1592, 1593,
1594, 1595, 1596, 1597, 1598, 1599,
1600, 1601, 1602, 1603, 1604, 1605,
1606, 1607, 1608, 1609, 1610, 1611,
1612, 1613, 1614, 1615, 1616, 1617,
1618, 1619, 1620, 1621, 1622, 1623,
1624, 1625, 1626, 1627, 1628, 1629,
1630, 1631, 1632, 1633, 1634, 1635,
1636, 1637, 1638, 1639, 1640, 1641,
1642, 1643, 1644, 1645, 1646, 1647,
1648, 1649, 1650, 1651, 1652, 1653,
1654, 1655, 1656, 1657, 1658, 1659,
1660, 1661, 1662, 1663, 1664, 1665,
1666, 1667, 1668, 1669, 1670, 1671,
1672, 1673, 1674, 1675, 1676, 1677,
1678, 1679, 1680, 1681, 1682, 1683,
1684, 1685, 1686, 1687, 1688, 1689,
1690, 1691, 1806, 1807, 1808, 1809,
1810, 1811, 1812, 1813, 1814, 1815,
1816, 1817, 1818, 1819, 1820, 1821

\DeclareTextCompositeCommand ...
..... 176, 1428, 1451,
1452, 1453, 1454, 1456, 1692, 1693,
1695, 1698, 1699, 1700, 1701, 1702,
1703, 1704, 1705, 1706, 1789, 11009

\DeclareTextFontCommand .....
..... w1, w15, w16, w17,
w18, w19, w20, w21, w22, w23,
w24, w29, w30, w31, w42, w140

\DeclareTextSymbol .....
..... 13, 1398, 1399, 1400, 1401,
1402, 1403, 1404, 1405, 1406, 1407,
1408, 1409, 1410, 1413, 1414, 1415,
1416, 1417, 1418, 1523, 1524, 1525,
1526, 1527, 1528, 1529, 1530, 1531,
1532, 1533, 1534, 1535, 1536, 1537,
1538, 1540, 1541, 1542, 1543, 1544,
1545, 1546, 1547, 1548, 1549, 1550,
1551, 1552, 1553, 1554, 1555, 1556,
1557, 1558, 1559, 1560, 1561, 1562,
1563, 1564, 1565, 1566, 1567, 1568,
1569, 1570, 1571, 1572, 1573, 1574,
1575, 1576, 1577, 1578, 1579, 1580,
1711, 1712, 1713, 1714, 1715, 1716,
1717, 1718, 1719, 1720, 1721, 1722,
1723, 1733, 1734, 1762, 1763, 1764,
1765, 1766, 1767, 1768, 1770, 1771,
1772, 1773, 1774, 1775, 1776, 1777,
1778, 1779, 1780, 1781, 1782, 1783,
1784, 1785, 1786, 1787, 1788, 1847,
1848, 1849, 1850, 1851, 1852, 1853,
1854, 1855, 1856, 1857, 1858, 1859,
1860, 1861, 1862, 1863, 1864, 1865,
1866, 1867, 1868, 1869, 1870, 1871,
1872, 1873, 1874, 1875, 1882, 1883,
1884, 1885, 1886, 1887, 1888, 1889,
1890, 1891, 1892, 1893, 1894, 1895,
1896, 1897, 1898, 1899, 1900, 1901,
1902, 1903, 1904, 1905, 1906, 1907,
1908, 1909, 1910, 1911, 1912, 1913,
1914, 1915, 1916, 1917, 1918, 1919,
1920, 1921, 1922, 1923, 1924, 1925,
1926, 1927, 1928, 1929, 1930, 1931,
1932, 1933, 1934, 1935, 1936, 1937,
1938, 1939, 1940, 1941, 1942, 1943,
1944, 1945, 1946, 1947, 1948, 1949,
1950, 1951, 1952, 1953, 1954, 1955,
1956, 1957, 1958, 1959, 1960, 1961,
11035, 11036, 11037, 11038, 11039,
11040, 11041, 11042, 11043, 11044,
11045, 11046, 11047, 11048, 11049,
11050, 11051, 11052, 11053, 11054,
11055, 11057, 11058, 11059, 11060,
11061, 11062, 11063, 11064, 11065,
11066, 11067, 11068, 11069, 11070,
11071, 11073, 11074, 11075, 11076,
11077, 11078, 11079, 11080, 11081,
11082, 11083, 11084, 11085, 11086,
11087, 11088, 11089, 11090, 11091,
11092, 11093, 11094, 11095, 11096,

```

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- l1097, l1098, l1099, l1100, l1101,
 l1102, l1103, l1104, l1105, l1106,
 l1107, l1108, l1109, l1110, l1111,
 l1112, l1113, l1114, l1115, l1116,
 l1117, l1118, l1119, l1120, l1121,
 l1122, l1123, l1124, l1125, l1126,
 l1127, l1128, l1129, l1130, l1131,
 l1132, l1133, l1134, l1135, l1136,
 l1137, l1138, l1139, l1140, l1141,
 l1142, l1143, l1144, l1145, l1146,
 l1147, l1148, l1149, l1150, l1151,
 l1152, l1153, l1155, l1156, l1168,
 l1169, l1170, l1171, l1172, l1173,
 l1174, l1175, l1176, l1177, l1178,
 x299, x300, x301, x302, x303,
 x304, x305, x306, x317, x318,
 x319, x320, x321, x322, x323,
 x324, x325, x326, x519, x520,
 x521, x522, x523, x524, x525, x526
 \DeclareTextSymbolDefault
 l185, l240, l241,
 l242, l243, l244, l245, l246, l247,
 l248, l249, l250, l251, l252, l253,
 l254, l255, l256, l257, l258, l259,
 l260, l261, l262, l263, l264, l265,
 l266, l267, l268, l269, l270, l271,
 l272, l273, l274, l275, l276, l277,
 l278, l280, l281, l291, x134,
 x136, x138, x144, x145, x146,
 x147, x148, x149, x150, x151,
 x152, x153, x154, x155, x156,
 x157, x158, x159, x160, x161,
 x162, x163, x164, x165, x166,
 x167, x168, x169, x170, x171,
 x172, x173, x174, x175, x176,
 x177, x178, x507, x508, x509,
 x510, x511, x512, x513, x514,
 x527, x528, x529, x530, x531,
 x532, x533, x534, x543, x544,
 x562, x563, x564, x565, x566,
 x567, x568, x570, x866, x867,
 x868, x869, x870, x871, x872,
 x873, x874, x875, x876, x877,
 x878, x879, x880, x881, x882,
 x883, x884, x885, x886, x887,
 x888, x889, x890, x891, x892,
 x893, x894, x895, x896, x897,
 x898, x899, x900, x901, x902,
 x903, x904, x905, x906, x907,
 x908, x909, x910, x911, x912,
 x913, x914, x915, x916, x917, x918
 \DeclareUnicodeAccent l1002,
 l1179, l1180, l1181, l1182, l1183,
 l1184, l1185, l1186, l1187, l1188,
 l1189, l1190, l1191, l1192, l1193
 \DeclareUnicodeCharacter Q280
 \DeclareUnicodeComposite
 l1007, 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, l1247, l1248,
 l1249, l1250, l1251, l1252, l1253,
 l1254, l1255, l1256, l1257, l1258,
 l1259, l1260, l1261, l1262, l1263,
 l1264, l1265, l1266, l1267, l1268,
 l1269, l1270, l1271, l1272, l1273,
 l1274, l1275, l1276, l1277, l1278,
 l1279, l1280, l1281, l1282, l1283,
 l1284, l1285, l1286, l1287, l1288,
 l1289, l1290, l1291, l1292, l1293,
 l1294, l1295, l1296, l1297, l1298,
 l1299, l1300, l1301, l1302, l1303,
 l1304, l1305, l1306, l1307, l1308,
 l1309, l1310, l1311, l1312, l1313,
 l1314, l1315, l1316, l1317, l1318,
 l1319, l1320, l1321, l1322, l1323,
 l1324, l1325, l1326, l1327, l1328,
 l1329, l1330, l1331, l1332, l1333,
 l1334, l1335, l1336, l1337, l1338,
 l1339, l1340, l1341, l1342, l1343,
 l1344, l1345, l1346, l1347, l1348,
 l1349, l1350, l1351, l1352, l1353,
 l1354, l1355, l1356, l1357, l1358,
 l1359, l1360, l1361, l1362, l1363,
 l1364, l1365, l1366, l1367, l1368,
 l1369, l1370, l1371, l1372, l1373,
 l1374, l1375, l1376, l1377, l1378,
 l1379, l1380, l1381, l1382, l1383,
 l1384, l1385, l1386, l1387, l1388,
 l1389, l1390, l1391, l1392, l1393,
 l1394, l1395, l1396, l1397, l1398,
 l1399, l1400, l1401, l1402, l1403,
 l1404, l1405, l1406, l1407, l1408,
 l1409, l1410, l1411, l1412, l1413,
 l1414, l1415, l1416, l1417, l1418,
 l1419, l1420, l1421, l1422, l1423,
 l1424, l1425, l1426, l1427, l1428,
 l1429, l1430, l1431, l1432, l1433,
 l1434, l1435, l1436, l1437, l1438
 \default@ds
 . . N214, N225, N261, N493, N495

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\default@family</code>	x846, B72, B73, B186, B187, B188, B189, D394, D397, E164, E165, M508, M510, M531, M533
..... o93, o125, o422, o436, o439, o464, o499, p93, p125, p422, p436, p439, p464, p499, Q370	
<code>\default@M</code> o100, o140, o143, o147, p100, p140, p143, p147, Q377	
<code>\default@extra</code> r10, r89	
<code>\default@series</code>	
..... o93, o126, o423, o437, o440, o461, o496, p93, p126, p423, p437, p440, p461, p496, Q370	
<code>\default@shape</code> o94, o127, o424, o438, o441, o459, o494, p94, p127, p424, p438, p441, p459, p494, Q371	
<code>\default@T</code> o134, o137, o147, o236, p134, p137, p147, p236	
<code>\defaultthyphenchar</code> b322, d403	
<code>\defaultscriptratio</code>	
..... o559, o566, p559, p566	
<code>\defaultscriptratio</code>	
..... o560, o566, p560, p566	
<code>\defaultskewchar</code> b323	
<code>\define@mathalphabet</code> r18, r131	
<code>\define@mathgroup</code> r19, r135	
<code>\define@newfont</code> o290, o299, p290, p299	
<code>\deg</code> B34	
<code>\delcode</code> s947	
<code>\delimiter</code> s800, s859, s930, s941	
<code>\delimiterfactor</code> b324	
<code>\delimitershortfall</code> b334	
<code>\Delta</code> u290	
<code>\delta</code> u263	
<code>\depth</code> D32, D35	
<code>\det</code> B30	
<code>\detokenize</code> l980, l1000, t238, x65, A51, N625, N626, N630	
<code>\development@branch@name</code>	
.. c11, c49, c50, c51, c57, c58, c59	
<code>\DH</code> l525, l1079, Q481	
<code>\dh</code> l535, l1085, Q481	
<code>\Diamond</code> t325	
<code>\diamond</code> u372	
<code>\diamondsuit</code> u324	
<code>\dim</code> B28	
<code>\dimen@</code> b41, b391, b392, b428, b429, b431, b433, g28, g29, i295, i300, l426, l427, l429, l430, l790, l791, o178, o180, o186, o199, o202, o206, o558, o559, o560, o564, p178, p180, p186, p199, p202, p206, p558, p559, p560, p564, q405, q406, q407, q408, q412, x123, x125, x844,	
	x846, B72, B73, B186, B187, B188, B189, D394, D397, E164, E165, M508, M510, M531, M533
<code>\dimen@i</code>	b41
<code>\dimen@ii</code> . b41, o182, o187, p182, p187	
<code>\dimendef</code> b42, b43, b44, b52, b91, P213	
<code>\dimenzero</code>	P213
<code>\dimexpr</code> l500, l516, l1161, l1164	
<code>\directlua</code>	a9, a12, a17, a20, a25, b65, b81, b105, b245, d19, d340, N609, P2, P12, P27, P204, P218, P243, P248, P252
<code>\disable_callback</code>	592, P749
<code>\discretionary</code> d401, d413, B205	
<code>\display</code>	B191, B195, B196
<code>\displaylines</code>	B190
<code>\displaymath</code>	B301
<code>\displaymath (environment)</code>	B299
<code>\displaystyle</code>	
..... u524, u527, u531, u533, B62, B197, B325, B328, B365, B390, B402, B430, B454, B457	
<code>\displaywidowpenalty</code>	b314
<code>\displaywidth</code> B197, B324, B377, B433	
<code>\div</code>	u375
<code>\DJ</code> l526, l1089, Q481	
<code>\dj</code> l536, l1090, Q481	
<code>\do</code>	a74, a75, a126, b13, b14, d44, f3, f7, f16, f26, k63, k66, k118, k121, k187, k259, k298, k391, k397, w90, A232, A253, A335, A345, A351, A357, D52, E220, E245, F31, F36, F83, F186, F188, F208, F211, F250, F294, F407, K16, K41, N109, N126, N232, N246, N258, N263, N283, N293, N536, N578, N658, N748, N1025, l65, l134
<code>\do@add@percent@to@temptokena</code> ..	
..... A52, A58, A76	
<code>\do@emfont@update</code> ... t232, t236, t256	
<code>\do@noligs</code>	A352, A357
<code>\do@subst@correction</code>	
..... o48, p48, q436, q491	
<code>\DocInput</code>	q8, u5, v5, O4
<code>\document</code>	88, k11, K40
<code>\document@default@language</code>	
..... k49, k50, M597, Q217	
<code>\document@select@group</code> ... s137, s236	
<code>\documentclass</code>	
..... q2, u2, v2, N300, N307, N334, N337, N460, N556, O2, P14	
<code>\documentstyle</code>	N305, N556

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\dont@add@percent@to@temptokena` .
 [A51](#), [A53](#), [A77](#)
`\doreset@version` [s114](#), [s119](#)
`\dospecials` .. [a74](#), [a126](#), [b13](#), [A232](#),
 [A253](#), [A335](#), [A345](#), [N658](#), [N748](#)
`\dot` [u517](#)
`\doteq` [u461](#)
`\dotfill` [b435](#), [d426](#), [d447](#)
`\dots` [l320](#), [l322](#)
`\doublehyphenemerits` [b317](#)
`\doublerulesep` [E287](#), [E314](#), [E338](#)
`\Downarrow` [u578](#)
`\downarrow` [u572](#)
`\downbracefill` [u530](#), [u549](#)
`\ds@` [N227](#), [N497](#)
`\dt@pfalse` [B192](#)
`\dt@ptrue` [B191](#)
`\dump` [Q548](#)
- E**
- `\E` [N665](#), [N668](#), [N695](#), [N755](#), [N758](#), [N786](#)
`\e@alloc` .. [b51](#), [b52](#), [b53](#), [b55](#), [b56](#),
 [b63](#), [b64](#), [b66](#), [b68](#), [b79](#), [b82](#),
 [b84](#), [b138](#), [P13](#), [P47](#), [P78](#), [P88](#),
 [P177](#), [P185](#), [P193](#), [P201](#), [Q12](#), [Q33](#)
`\e@alloc@attribute@count`
 .. [P66](#), [P74](#), [P75](#), [P79](#), [P81](#), [P224](#)
`\e@alloc@bytecode@count` ... [P70](#),
 [P189](#), [P190](#), [P194](#), [P196](#), [P240](#)
`\e@alloc@ccodetable@count`
 .. [P67](#), [P84](#), [P85](#), [P89](#), [P92](#), [P228](#)
`\e@alloc@chardef`
 [b60](#), [b102](#), [b210](#), [b211](#),
 [P46](#), [P177](#), [P185](#), [P193](#), [P201](#), [Q12](#)
`\e@alloc@intercharclass@top` ... [Q21](#)
`\e@alloc@luachunk@count` ... [P71](#),
 [P197](#), [P198](#), [P202](#), [P206](#), [P242](#)
`\e@alloc@luafunction@count`
 [P68](#), [P173](#),
 [P174](#), [P178](#), [P180](#), [P234](#), [P236](#)
`\e@alloc@top` [b55](#),
 [b63](#), [b102](#), [b188](#), [b248](#), [P45](#),
 [P79](#), [P178](#), [P186](#), [P194](#), [P202](#), [Q12](#)
`\e@alloc@whatsit@count` [P69](#),
 [P181](#), [P182](#), [P186](#), [P188](#), [P238](#)
`\e@ch@ck` [b142](#), [b152](#), [P49](#), [P53](#)
`\e@insert@top` . [b246](#), [b248](#), [b265](#), [b280](#)
`\e@mathgroup@top` [b79](#), [b124](#), [s56](#), [s145](#)
`\egroup` [b372](#)
`\eject` [b402](#)
`\ell` [u303](#)
`\em` [t225](#), [t252](#), [w42](#)
`\em@currfont` [t231](#), [t242](#)
`\em@force` [t240](#), [t245](#), [t248](#), [t258](#)
- `\emergencystretch` [L59](#), [L65](#)
`\emfontdeclare@clist` [t221](#),
 [t223](#), [t227](#), [t232](#), [t237](#), [t243](#), [t254](#)
`\emforce` [t261](#), [t221](#), [t248](#), [t257](#)
`\eminnershapes` [t229](#), [t235](#), [t252](#)
`\emph` [w42](#)
`\empty` [b370](#)
`\empty@sfcnt`
 [q444](#), [q445](#), [q446](#), [q460](#), [q465](#), [q517](#)
`\emptyset` [u310](#)
`\emreset` [t262](#), [t221](#), [t224](#), [t255](#)
`\emrest` [t224](#)
`\enc@update` [o221](#), [o223](#), [o239](#),
 [o242](#), [p221](#), [p223](#), [p239](#), [p242](#), [q129](#)
`\encodingdefault`
 [l973](#), [l1472](#), [l1498](#), [l1499](#),
 [s237](#), [t312](#), [u48](#), [P247](#), [P262](#), [P270](#)
`\end` . [a69](#), [d8](#), [d362](#), [g200](#), [q9](#), [u6](#), [v6](#),
 [A92](#), [A118](#), [A211](#), [A212](#), [B411](#),
 [B420](#), [C112](#), [H15](#), [H17](#), [N673](#),
 [N677](#), [N683](#), [N763](#), [N767](#), [N773](#), [O5](#)
`\end@dblfloat` [I205](#)
`\end@float` [I189](#), [I227](#), [I243](#), [I359](#)
`\endarray` [E159](#)
`\endcenter` [A173](#)
`\enddisplaymath` [B302](#)
`\enddocument` [A8](#)
`\endenumerate` [C240](#)
`\endeqnarray` [B333](#), [B364](#)
`\endequation` [B305](#)
`\endfilecontents` [N560](#)
`\endflushleft` [A203](#)
`\endflushright` [A205](#)
`\endgraf` [b367](#)
`\EndIncludeInRelease`
 [a22](#), [a50](#), [a306](#), [a316](#), [b87](#), [b101](#),
 [b118](#), [b123](#), [b133](#), [b137](#), [b147](#),
 [b150](#), [b167](#), [b181](#), [b185](#), [b219](#),
 [b224](#), [b277](#), [b289](#), [b480](#), [b487](#),
 [b534](#), [b539](#), [c97](#), [c103](#), [d273](#),
 [d299](#), [d302](#), [d318](#), [d351](#), [d360](#),
 [d411](#), [d415](#), [d440](#), [d461](#), [i20](#), [i30](#),
 [i64](#), [i77](#), [i125](#), [i139](#), [i151](#), [i162](#),
 [i179](#), [i191](#), [i256](#), [i278](#), [i344](#), [i348](#),
 [i366](#), [i373](#), [k69](#), [k123](#), [k164](#),
 [k176](#), [k269](#), [k308](#), [k322](#), [k330](#),
 [k345](#), [k355](#), [l103](#), [l123](#), [l168](#),
 [l175](#), [l330](#), [l352](#), [l367](#), [l375](#), [m28](#),
 [m33](#), [m89](#), [m100](#), [m142](#), [m148](#),
 [m168](#), [m171](#), [n10](#), [n14](#), [o195](#),
 [o212](#), [o365](#), [o370](#), [o405](#), [o410](#),
 [o427](#), [o445](#), [o484](#), [o517](#), [o633](#),
 [o645](#), [p195](#), [p212](#), [p365](#), [p370](#),
 [p405](#), [p410](#), [p427](#), [p445](#), [p484](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- p517, p633, p645, q503, q509,
 r21, r143, s77, s105, s168, s198,
 s622, s664, s677, s684, s831,
 s887, s936, s944, t172, t214,
 t251, t263, t270, u78, u88, u103,
 u109, u625, u632, w33, w40,
 x41, x55, x536, x723, z48, z59,
 A71, A84, A141, A147, A160,
 A167, A191, A199, A237, A258,
 A299, A318, A340, A348, B86,
 B95, B117, B124, B143, B147,
 B162, B170, B233, B241, B270,
 B297, B393, B405, B414, B423,
 C132, C137, D13, D22, D79,
 D86, D143, D150, D198, D206,
 D257, D275, D343, D348, D371,
 D378, E144, E152, F286, F327,
 F437, F454, H19, H29, H167,
 H172, H203, H225, L39, L46,
 M53, M62, M178, M196, M365,
 M370, M418, M464, M650,
 M708, M811, M830, M893,
 M911, M953, M974, M1216,
 M1385, M1467, M1561, M1683,
 M1810, M1929, M1957, M2177,
 M2195, M2242, M2286, N72,
 N83, N121, N138, N166, N179,
 N288, N297, N408, N436, N698,
 N788, P221, P244, P267, P271,
 Q15, Q19, Q37, Q55, Q65,
 Q72, Q80, Q131, Q155, Q223,
 Q227, Q336, Q383, I104, I172,
 I231, I246, I293, I306, I391, I396
 \EndIncludeRelease c73
 \enditemize C251
 \endline b367, B175
 \endlinechar a92, a93, a94, a204, d22,
 d24, d29, k365, N187, N188, N189
 \endlist C98, C240, C251
 \endlrbox D110
 \endmath B300
 \endminipage D309
 \endpicture F17
 \endsloppypar L63
 \endtabbing E72
 \endtabular E159
 \endtabular* E159
 \endtrivlist ... A173, A203, A205,
 A261, B435, C100, C101, E73, G39
 \endverbatim A260, A297
 \enlargethispage M1855
 \enlargethispage* M1855
 \enskip i375
 \enspace i361, i371
 \ensuremath ... m144, B366, I381, I389
 \enumerate C231
 enumerate (environment) C231
 environments:
 center A172
 displaymath B299
 enumerate C231
 eqnarray B311, B436
 eqnarray* B362
 equation B303, B424
 filecontents 554
 flushleft A202
 flushright A204
 itemize C242
 lrbox 379
 math B299
 minipage 380
 sloppypar L62
 thebibliography 474
 verbatim* A294
 \epsilon u264
 \eqnarray B316, B363
 eqnarray (environment) ... B311, B436
 eqnarray* (environment) B362
 \eqno B305
 \equation B304
 equation (environment) ... B303, B424
 \equiv u440
 \err@rel@i r12, r99, r132, r136
 \errhelp a217,
 c31, g39, g66, O12, Q237, Q537
 \errmessage a4, a58, a222,
 b164, b178, b293, c32, g47, g72,
 o456, o491, p456, p491, q379,
 q479, r65, O16, P61, Q49, Q239
 \error@fontshape
 o417, o432, o457, o492, p417,
 p432, p457, p492, q107, q481, s222
 \errorcontextlines b327,
 b454, b470, b485, b498, b515, g163
 \errorstopmode b443, Q547
 \escapechar
 d101, d144, d148, d156, k228,
 o302, o526, p302, p526, q183,
 s58, s86, s147, s177, s221, P203
 \et@xmaxfam P20, P24, P28, P36
 \et@xmaxregs P27,
 P29, P30, P31, P32, P33, P34, P35
 \eta u266
 \etatcatcode P783
 \eTeXversion a57
 \evensidemargin ... M73, M613, M672
 \every@math@size o42, p42, q189, q201
 \everycr b423, B192, B195, B324, B451

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\everydisplay`
 o280, [o281](#), o286, p280, [p281](#), p286
 - `\everyjob` c37, c42,
 c47, s241, P207, P208, P249,
 P250, Q312, Q506, Q507, Q509
 - `\everymath`
 o279, [o281](#), o284, p279, [p281](#), p284
 - `\everypar`
 69, k41, k99, o579, o592, o639,
 p579, p592, p639, A87, A234,
 A256, C129, C131, C135, C136,
 C180, C197, D247, D268, E69,
 H48, H96, H107, H127, H136,
 M165, M192, M1151, M1317, I187
 - `\execute@size@function`
 q316, q344, q358, q375
 - `\ExecuteOptions`
 q57, q70, x756, x793, [N276](#)
 - `\exhyphenpenalty` b309, b396
 - `\exists` u316
 - `\exp` B31
 - `\expand@font@defaults` [t69](#),
 t79, t91, t121, t164, t186, x26, x104
 - `\external@font`
 q84, q87, q98, q102, q104, q345,
 q359, q421, q455, q523, q525, q527
 - `\extra@def` r9, [r84](#)
 - `\extracolsep` [E155](#)
 - `\extract@alph@from@version`
 o532, [o538](#), [p532](#), [p538](#), s151, s182
 - `\extract@default@composite`
 I1012, I1019
 - `\extract@default@composite@a` ...
 I1021, I1025
 - `\extract@default@composite@b` ...
 I1023, I1027
 - `\extract@font` o313, p313, [q81](#)
 - `\extract@fontinfo` q312, q319
 - `\extract@range@fontinfo`
 [q329](#), q336, q355, q388
 - `\extract@size@fn` [q304](#), q326
 - `\extrafloats` [b152](#), [b189](#), b263
- F**
- `\f@baselineskip` I1162, o252, o259,
 o443, p252, p259, p443, q119,
 q136, q140, q155, q169, q180, q194
 - `\f@depth` [M345](#), I291
 - `\f@encoding` I178,
 o215, o234, o237, o238, o240,
 o261, o293, o298, o317, o319,
 o321, o326, o328, o359, o421,
 o453, o488, [p215](#), p234, p237,
 p238, p240, p261, p293, p298,
 p317, p319, p321, p326, p328,
 p359, p421, p453, p488, q91,
 q261, q471, s207, P247, P262, P270
 - `\f@family` [o243](#),
 o247, o255, o294, o298, o317,
 o319, o321, o326, o328, o360,
 o439, o464, o499, [p243](#), p247,
 p255, p294, p298, p317, p319,
 p321, p326, p328, p360, p439,
 p464, p499, q91, s207, t81, t82,
 t83, t93, t94, t95, t138, x23,
 x27, x29, x31, x84, x87, x101,
 x105, x107, x109, x114, x121,
 x810, x813, x827, x836, x842, x1057
 - `\f@linespread`
 [o255](#), [p255](#), q118, q137,
 q138, q141, q149, q152, q163, q166
 - `\f@series` j14, [o243](#),
 o248, o256, o295, o298, o440,
 o461, o496, [p243](#), p248, p256,
 p295, p298, p440, p461, p496,
 t131, t144, t146, t148, t149, x7, x48
 - `\f@shape` [o243](#), o249, o257, o296, o298,
 o441, o459, o494, [p243](#), p249,
 p257, p296, p298, [p441](#), p459, p494
 - `\f@size` I180, I1161, o52, o252,
 o258, o297, o442, o481, o515,
 o557, o558, o561, o562, p52,
 p252, p258, p297, p442, p481,
 p515, p557, p558, p561, p562,
 q119, q121, q134, q154, q169,
 q172, q175, q180, q187, q194,
 q206, q209, q215, q221, q238,
 q239, q242, q247, q313, q320,
 q339, q341, q356, q407, q409,
 q411, q427, q428, q433, q447,
 q459, q464, q476, q484, q489,
 q515, q529, t231, t240, x123, x844
 - `\f@user@size` .. q427, q432, q476, q489
 - `\fam` b98, o15, p15, P20, P24, P36
 - `\famdef@ult` t75, t165, t166, t167
 - `\familydefault`
 I1499, s238, t75, t313, [u111](#)
 - `\fbox` 379, [D128](#), D141, D148
 - `\fboxrule` [D126](#), D162, D165,
 D171, D173, D180, D181, Q84
 - `\fboxsep` [D126](#), D132,
 D161, D166, D176, D178, Q83
 - `\filbreak` [b400](#)
 - `\filecontents`
 ... N569, N572, N575, N585,
 N606, N700, N722, N810, N1025
 - `\filecontents@checkdir`
 N591, N593, N607, N707

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=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \filec@ntents@force [N587](#), [N703](#)
- \filec@ntents@noheader . . [N589](#), [N705](#)
- \filec@ntents@nosearch . . [N590](#), [N706](#)
- \filec@ntents@opt
- [N572](#), [N575](#), [N577](#), [N702](#)
- \filec@ntents@overwrite . . [N588](#), [N704](#)
- \filec@ntents@where
- [N592](#), [N594](#), [N619](#), [N708](#)
- \filecontents [N560](#)
- filecontents (environment) [554](#)
- \filename@area . . [a246](#), [a252](#), [a259](#),
- [a265](#), [a272](#), [a278](#), [a285](#), [k341](#),
- [k352](#), [k376](#), [k379](#), [k393](#), [k405](#), [k407](#)
- \filename@base . . [a295](#), [a302](#), [a315](#),
- [k341](#), [k352](#), [k376](#), [k379](#), [k400](#), [k405](#)
- \filename@dot [a313](#), [a319](#)
- \filename@dots [a297](#), [a299](#), [a304](#)
- \filename@ext
- . [a294](#), [a301](#), [a311](#), [a313](#), [k342](#),
- [k353](#), [k372](#), [k373](#), [k376](#), [k379](#), [k401](#)
- \filename@parse [1](#), [6](#),
- [a110](#), [a242](#), [k339](#), [k350](#), [k371](#), [k398](#)
- \filename@path . . [a247](#), [a248](#), [a253](#),
- [a260](#), [a261](#), [a266](#), [a273](#), [a274](#), [a279](#)
- \filename@simple [a250](#), [a263](#),
- [a276](#), [a286](#), [a290](#), [a292](#), [a307](#), [a309](#)
- \fill [i354](#)
- \finalhyphendemerits [b318](#)
- \finph@nt [B104](#),
- [B106](#), [B110](#), [B111](#), [B119](#), [B120](#)
- \finsm@sh [B134](#),
- [B136](#), [B140](#), [B141](#), [B145](#), [B146](#)
- \firstmark . . [L51](#), [M647](#), [M706](#), [M2217](#)
- \fix@penalty [w101](#)
- \fixed@sfcnt [q519](#), [q520](#), [q521](#)
- \fl@trace [M240](#), [M267](#),
- [M323](#), [M351](#), [M358](#), [M379](#),
- [M426](#), [M472](#), [M525](#), [M540](#),
- [M541](#), [M542](#), [M543](#), [M554](#),
- [M555](#), [M556](#), [M557](#), [M558](#),
- [M568](#), [M781](#), [M800](#), [M819](#),
- [M837](#), [M839](#), [M978](#), [M982](#),
- [M994](#), [M995](#), [M996](#), [M997](#),
- [M1003](#), [M1006](#), [M1014](#), [M1018](#),
- [M1029](#), [M1034](#), [M1039](#), [M1040](#),
- [M1041](#), [M1042](#), [M1049](#), [M1052](#),
- [M1060](#), [M1071](#), [M1077](#), [M1082](#),
- [M1087](#), [M1093](#), [M1094](#), [M1099](#),
- [M1104](#), [M1105](#), [M1106](#), [M1114](#),
- [M1118](#), [M1123](#), [M1127](#), [M1132](#),
- [M1143](#), [M1144](#), [M1146](#), [M1164](#),
- [M1173](#), [M1179](#), [M1188](#), [M1191](#),
- [M1197](#), [M1207](#), [M1211](#), [M1221](#),
- [M1227](#), [M1233](#), [M1239](#), [M1246](#),
- [M1248](#), [M1254](#), [M1259](#), [M1261](#),
- [M1263](#), [M1271](#), [M1276](#), [M1282](#),
- [M1287](#), [M1293](#), [M1307](#), [M1308](#),
- [M1311](#), [M1332](#), [M1341](#), [M1347](#),
- [M1356](#), [M1359](#), [M1366](#), [M1376](#),
- [M1380](#), [M1392](#), [M1398](#), [M1403](#),
- [M1408](#), [M1412](#), [M1416](#), [M1417](#),
- [M1424](#), [M1429](#), [M1433](#), [M1440](#),
- [M1449](#), [M1453](#), [M1457](#), [M1458](#),
- [M1462](#), [M1463](#), [M1473](#), [M1479](#),
- [M1485](#), [M1491](#), [M1495](#), [M1501](#),
- [M1503](#), [M1511](#), [M1516](#), [M1521](#),
- [M1529](#), [M1538](#), [M1543](#), [M1548](#),
- [M1550](#), [M1555](#), [M1557](#), [M1568](#),
- [M1574](#), [M1584](#), [M1590](#), [M1594](#),
- [M1595](#), [M1600](#), [M1601](#), [M1607](#),
- [M1610](#), [M1611](#), [M1612](#), [M1619](#),
- [M1620](#), [M1621](#), [M1629](#), [M1634](#),
- [M1646](#), [M1647](#), [M1654](#), [M1657](#),
- [M1665](#), [M1669](#), [M1673](#), [M1674](#),
- [M1678](#), [M1679](#), [M1689](#), [M1695](#),
- [M1705](#), [M1711](#), [M1715](#), [M1716](#),
- [M1722](#), [M1723](#), [M1730](#), [M1733](#),
- [M1734](#), [M1735](#), [M1743](#), [M1744](#),
- [M1745](#), [M1754](#), [M1759](#), [M1772](#),
- [M1774](#), [M1781](#), [M1784](#), [M1793](#),
- [M1797](#), [M1801](#), [M1802](#), [M1806](#),
- [M1807](#), [M1859](#), [M1864](#), [M1870](#),
- [M1880](#), [M1887](#), [M1897](#), [M1993](#),
- [M2006](#), [M2007](#), [M2011](#), [M2014](#),
- [M2016](#), [M2019](#), [M2022](#), [M2024](#),
- [M2065](#), [M2072](#), [M2077](#), [M2083](#),
- [M2088](#), [M2092](#), [M2098](#), [M2106](#),
- [M2108](#), [M2115](#), [M2120](#), [M2125](#),
- [M2127](#), [M2133](#), [M2135](#), [M2142](#),
- [M2171](#), [M2173](#), [M2188](#), [M2190](#),
- [M2204](#), [M2229](#), [M2233](#), [M2238](#),
- [M2250](#), [M2267](#), [M2272](#), [M2280](#)
- \fl@tracemessage [M1897](#)
- \fl@traceval [M1897](#)
- \flat [u320](#)
- \float@count . . [b51](#), [b52](#), [b53](#), [b62](#),
- [b188](#), [b205](#), [b210](#), [b212](#), [b213](#), [b222](#)
- \floatingpenalty [l412](#)
- \floatpagefraction [M2298](#), [l278](#)
- \floatsep [M726](#), [M744](#),
- [M751](#), [M2104](#), [M2154](#), [M2303](#)
- \flushbottom [L55](#)
- \flushleft [A202](#)
- flushleft (environment) [A202](#)
- \flushright [A204](#)
- flushright (environment) [A204](#)
- \fmtname [c1](#), [c38](#),
- [c40](#), [c43](#), [c45](#), [c48](#), [c56](#), [N343](#), [N347](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdccl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \fmtversion
... [c1](#), [c19](#), [c38](#), [c40](#), [c43](#), [c45](#),
[c48](#), [c56](#), [c84](#), [g2](#), [t170](#), [E1](#), [F1](#),
[M4](#), [N360](#), [N363](#), [P263](#), [Q490](#), [Q516](#)
 - \fmtversion@topatch [Q488](#),
[Q490](#), [Q502](#), [Q503](#), [Q515](#), [Q523](#)
 - \fnsymbol [145](#), [m107](#)
 - \font . [b428](#), [b433](#), [d402](#), [d405](#), [l297](#),
[l298](#), [l299](#), [l436](#), [l443](#), [l793](#), [l800](#),
[l985](#), [l986](#), [l1010](#), [l1158](#), [o45](#),
[o51](#), [o53](#), [p45](#), [p51](#), [p53](#), [q84](#),
[t228](#), [t261](#), [t267](#), [t293](#), [v8](#), [v9](#),
[v10](#), [w85](#), [x6](#), [x47](#), [x546](#), [x553](#), [A255](#)
 - \font@info [q99](#), [q319](#), [q388](#), [q393](#)
 - \font@name [l179](#), [l182](#), [o50](#),
[o158](#), [o160](#), [o289](#), [o304](#), [o480](#),
[o514](#), [p50](#), [p158](#), [p160](#), [p289](#),
[p304](#), [p480](#), [p514](#), [q84](#), [q88](#),
[q90](#), [q105](#), [q120](#), [q123](#), [q126](#),
[q284](#), [q285](#), [q286](#), [q287](#), [q288](#), [q293](#)
 - \font@submax [q395](#), [q424](#),
[q425](#), [A22](#), [A24](#), [Q229](#), [Q231](#), [Q240](#)
 - \fontdimen [b428](#), [b433](#),
[l297](#), [l298](#), [l299](#), [l436](#), [l443](#), [l793](#),
[l800](#), [t228](#), [t261](#), [t267](#), [w85](#), [x6](#),
[x47](#), [x546](#), [x553](#), [F38](#), [F40](#), [F365](#)
 - \fontencoding
[l1498](#), [o215](#), [o246](#), [p215](#), [p246](#),
[s237](#), [u16](#), [u23](#), [u65](#), [u72](#), [u81](#),
[u83](#), [x36](#), [P246](#), [P247](#), [P269](#), [P270](#)
 - \fontfamily [o243](#),
[p243](#), [s238](#), [t6](#), [t9](#), [t12](#), [t127](#),
[t197](#), [t200](#), [t203](#), [t336](#), [u56](#),
[u67](#), [u74](#), [u85](#), [x28](#), [x30](#), [x32](#),
[x34](#), [x106](#), [x108](#), [x110](#), [x112](#), [x829](#)
 - \fontname [l986](#), [o53](#), [p53](#)
 - \fontseries [o243](#), [p243](#), [s239](#),
[t15](#), [t18](#), [t80](#), [t92](#), [t191](#), [t194](#), [t337](#)
 - \fontshape . [l446](#), [l803](#), [o243](#), [p243](#),
[s240](#), [t21](#), [t24](#), [t27](#), [t30](#), [t338](#), [x556](#)
 - \fontsize [j6](#),
[l302](#), [l328](#), [l360](#), [l1160](#), [l1196](#),
[o43](#), [o253](#), [p43](#), [p253](#), [t299](#),
[t339](#), [x125](#), [x588](#), [x846](#), [I381](#), [I389](#)
 - \fontsubfuzz [q395](#), [q429](#), [A22](#)
 - \footins [M314](#), [M315](#), [M316](#),
[M317](#), [M375](#), [M422](#), [M482](#),
[M490](#), [M494](#), [M517](#), [I366](#), [I408](#)
 - \footnote [I399](#)
 - \footnotemark [H10](#), [I421](#)
 - \footnoterule [D315](#), [M493](#), [I370](#)
 - \footnotesep ... [D334](#), [I398](#), [I411](#), [I419](#)
 - \footnotesize [D327](#), [I409](#)
 - \footnotetext [H12](#), [I438](#)
 - \footskip [M77](#), [M637](#), [M696](#)
 - \forall [u315](#)
 - \fps@dbl [I34](#)
 - \frac [B308](#)
 - \frame [D112](#), [D188](#)
 - \framebox [379](#), [D135](#)
 - \frenchspacing [b353](#), [d427](#),
[d448](#), [k44](#), [k102](#), [A260](#), [A296](#), [A350](#)
 - \frown [u443](#)
 - \frozen@everydisplay
..... [o279](#), [o285](#), [p279](#), [p285](#)
 - \frozen@everymath
..... [o279](#), [o283](#), [p279](#), [p283](#)
 - \fussy [L64](#)
 - \futurelet [d368](#),
[d382](#), [i320](#), [i328](#), [w83](#), [B210](#), [E335](#)
- ## G
- \g@addto@macro [N504](#), [N510](#), [N514](#), [N515](#)
 - \G@refundefinedfalse [z5](#)
 - \G@refundefinedtrue [z3](#), [z12](#), [K21](#), [K44](#)
 - \Gamma [u289](#)
 - \gamma [u262](#)
 - \gcd [B33](#)
 - \ge [u409](#), [u411](#)
 - \gen@sfcnt [q456](#), [q457](#), [q458](#)
 - \genb@sfcnt [q461](#), [q462](#), [q463](#)
 - \genb@x [q464](#), [q466](#)
 - \genb@y [q466](#)
 - \GenericError [g18](#), [g85](#), [g111](#), [g137](#), [g62](#)
 - \GenericInfo
... [c85](#), [c88](#), [c93](#), [g4](#), [g104](#), [g130](#),
[g155](#), [q31](#), [q34](#), [q39](#), [q75](#), [N951](#)
 - \GenericWarning [g11](#),
[g94](#), [g120](#), [g146](#), [q42](#), [q47](#), [q50](#), [q78](#)
 - \geq [u407](#), [u409](#)
 - \get@cdp [s356](#), [s364](#), [s397](#)
 - \get@external@font [q83](#), [q96](#), [q490](#)
 - \getanddefine@fonts
..... [o527](#), [o545](#), [p527](#), [p545](#),
[q274](#), [s59](#), [s87](#), [s132](#), [s148](#), [s178](#),
[s263](#), [s327](#), [s361](#), [s363](#), [s380](#),
[s503](#), [s504](#), [s536](#), [s537](#), [s1043](#), [s1044](#)
 - \GetFileInfo [u3](#)
 - \gets [u431](#), [u433](#)
 - \gg [u425](#)
 - \glb@currsize [k39](#), [k97](#), [o276](#),
[p276](#), [q171](#), [q206](#), [q210](#), [q216](#), [q239](#)
 - \glb@settings
..... [o277](#), [p277](#), [q171](#), [q218](#), [q249](#)
 - \globaldefs [o528](#),
[p528](#), [q185](#), [s60](#), [s89](#), [s149](#), [s180](#)
 - \glossary [472](#), [H176](#),
[J23](#), [J35](#), [L24](#), [L32](#), [M621](#), [M680](#)

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=ltlgos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\glossaryentry` J32
`\goodbreak` [b400](#), d428, d449
`\grave` u509
`\group@elt` s35,
 s261, s298, s299, [s320](#), s324, s1075
`\group@list` s265, s305, [s318](#),
 s323, s324, s353, s579, s627,
 s688, s777, s780, s836, s839,
 s892, s895, s960, s963, s1030, s1081
`\guillemetleft` l537, l767, l1055
`\guillemetright` l538, l768, l1071
`\guillemotleft` l540, l770, l1057
`\guillemotright` l541, l771, l1073
`\guilsinglleft` l542, l1125
`\guilsinglright` l543, l1126
- ## H
- `\H` . g24, l230, l385, l466, l595, l603,
 l622, l630, l747, l1188, l1327,
 l1328, l1355, l1356, x192, x209
`\h@false` B81
`\h@true` B82, B83
`\halign` . [b423](#), [B127](#), [B197](#), [B324](#), [B451](#)
`\hangindent` H139
`\hat` u515
`\hb@xt@` [b438](#), [d14](#),
 l422, [B197](#), [B329](#), [B375](#), [B390](#),
 [B402](#), [B429](#), [B459](#), [D44](#), [D59](#),
 [D160](#), [D402](#), [D406](#), [D407](#), [E37](#),
 [F13](#), [F23](#), [F32](#), [F123](#), [F157](#),
 [F160](#), [F163](#), [F165](#), [F167](#), [F238](#),
 [F279](#), [F322](#), [F417](#), [H197](#), [H220](#),
 [H227](#), [M630](#), [M640](#), [M689](#),
 [M699](#), [M1843](#), [M2223](#), [M2224](#),
 [M2228](#), [M2255](#), [M2256](#), [M2262](#)
`\hbadness` [b305](#), o582, o589,
 o624, o643, p582, p589, p624, p643
`\hbar` u327
`\headheight` M75, M626, M685
`\headsep` M76, M635, M694
`\heartsuit` u325
`\height` l1164, D31, D34
`\hexnumber@` s600, s608, s643,
 s651, s672, s682, s708, s716,
 s724, s733, s736, s745, s746,
 s789, s797, s848, s856, s904,
 s912, s930, s931, s941, s942,
 s947, s973, s981, s986, s988, [t303](#)
`\hfuzz` b328,
 o590, p590, L60, L61, L67, L68
`\hgl@` b393, b394
`\hglue` [b390](#)
`\hideoutput` [b488](#)
`\hideskip` [b296](#), [b414](#)
- `\hidewidth` [b414](#), l327,
 l329, l358, l362, l390, l391, l394,
 l397, l473, l474, l478, l481, l483,
 l486, l498, l503, l519, l754, l755,
 l758, l761, l828, l831, l1195, l1197
`\hline` [E334](#), [E337](#)
`\hmode@bgroup` l67, [l75](#), l327,
 l356, l390, l396, l424, l435, l442,
 l473, l480, l483, l485, l493, l509,
 l724, l754, l760, l792, l799, l827,
 l830, l876, l1195, w7, x545, x552
`\hmode@start@before@group`
 [l68](#), [l151](#), [l153](#), [l159](#), [l184](#)
`\hom` B29
`\hookleftarrow` u472
`\hookrightarrow` u470
`\hphantom` [B75](#)
`\hrule` [b391](#),
 b435, i296, i304, l289, l294,
 u332, u610, D118, D123, D171,
 D181, [E335](#), [E352](#), [F281](#), [F324](#), [I371](#)
`\hrulefill` [b435](#), d429, d450
`\hspace` [i350](#)
`\Hwithstroke` l491, l1155
`\hwithstroke` l507, l1156
`\hyphenation` [l205](#)
`\hyphenchar` d395, d402, d405, d412, A255
`\hyphenpenalty`
 b308, o596, o628, p596, p628
- ## I
- `\I` [b359](#), N691, N782, N802, Q188, Q451
`\i` l247, l402,
 l449, l450, l451, l452, l453, l454,
 l544, l581, l582, l674, l676, l678,
 l680, l772, l1091, l1237, l1239,
 l1241, l1243, l1294, l1297, l1300,
 l1303, l1373, Q192, Q455, Q462
`\ialign` [b423](#), b425,
 u329, u451, u522, u525, u529,
 u532, B155, B157, B176, E179, F52
`\IeC` Q253, Q257, Q364
`\if@afterindent` [H124](#), [H131](#)
`\if@compatibility` [N2](#), [N302](#)
`\if@endpe` [A138](#), [C138](#)
`\if@eqnsw` [B311](#), [B360](#)
`\if@fcolmade` ... [M95](#), M264, M394,
 M403, M441, M451, M779,
 M799, M817, M846, M926,
 M2170, M2187, M2237, M2277
`\if@filesw`
 [k7](#), k34, k92, k180, k192, k199,
 k208, A14, A28, H153, K4, K8,
 K19, K28, K36, K43, N617, 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=ltlogss.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
 o=ltfssbas.dtx, p=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\if@firstamp` [E227](#)
`\if@firstcolumn`
 ... [M95](#), [M246](#), [M279](#), [M396](#),
 [M444](#), [M1815](#), [M2201](#), [M2246](#)
`\if@ignore` [A4](#), [A139](#)
`\if@includeinrelease` [c68](#), [c71](#), [c98](#), [d417](#)
`\if@inlabel` [C28](#), [C65](#),
 [C102](#), [C160](#), [C183](#), [M161](#), [M188](#)
`\if@insert` [M95](#), [M1057](#),
 [M1169](#), [M1203](#), [M1337](#), [M1372](#),
 [M1446](#), [M1535](#), [M1662](#), [M1790](#)
`\if@minipage` [i209](#), [i226](#), [i261](#),
 [A219](#), [A241](#), [C149](#), [D278](#), [E67](#), [I20](#)
`\if@mparswitch` [M95](#), [M1817](#)
`\if@multiplelabels` [z31](#)
`\if@negarg` [F56](#), [F78](#), [F92](#), [F131](#)
`\if@newlist` [A261](#),
 [C29](#), [C33](#), [C69](#), [C78](#), [C106](#),
 [C166](#), [M599](#), [M644](#), [M657](#), [M703](#)
`\if@nmbrlist` [C33](#), [C201](#)
`\if@no@font@opt` [r16](#), [r110](#), [r129](#)
`\if@nobreak`
 [i94](#), [i121](#), [i228](#), [i263](#), [k131](#), [k143](#),
 [C167](#), [C192](#), [D241](#), [D262](#), [H47](#),
 [H128](#), [L29](#), [L37](#), [M165](#), [M192](#),
 [M335](#), [M1148](#), [M1314](#), [I180](#), [I349](#)
`\if@noitemarg` [C32](#), [C199](#)
`\if@noparitem` [C30](#), [C157](#)
`\if@noparlist` [C31](#), [C114](#)
`\if@noskipsec` [i121](#), [C58](#), [D242](#), [D263](#),
 [H38](#), [H40](#), [H97](#), [M155](#), [M182](#), [I350](#)
`\if@ovb` . [F213](#), [F266](#), [F271](#), [F310](#), [F315](#)
`\if@ovhline` [F245](#), [F281](#), [F291](#)
`\if@ovl` . [F213](#), [F264](#), [F283](#), [F306](#), [F325](#)
`\if@ovr` . [F213](#), [F263](#), [F280](#), [F305](#), [F323](#)
`\if@ovt` . [F213](#), [F265](#), [F276](#), [F309](#), [F319](#)
`\if@ovvline` [F245](#), [F274](#), [F290](#)
`\if@partsw` [k7](#), [k184](#)
`\if@pboxsw` [D233](#), [D336](#)
`\if@reversemargin` [M101](#), [M1820](#)
`\if@reversemarginpar` [M95](#)
`\if@rjfield` [E19](#), [E33](#)
`\if@specialpage` ... [M95](#), [M606](#), [M664](#)
`\if@tempswa` ... [a78](#), [a79](#), [a80](#), [b259](#),
 [e9](#), [k190](#), [o63](#), [o622](#), [p63](#), [p622](#),
 [s286](#), [s341](#), [s405](#), [s486](#), [s1074](#),
 [A30](#), [A226](#), [A247](#), [K52](#), [M990](#),
 [M1026](#), [M1626](#), [M1751](#), [N648](#), [N738](#)
`\if@test` [M12](#), [M13](#), [M887](#),
 [M906](#), [M946](#), [M968](#), [M1032](#),
 [M1116](#), [M1125](#), [M1274](#), [M1285](#),
 [M1427](#), [M1514](#), [M1632](#), [M1757](#)
`\if@twocolumn` [k24](#), [k81](#), [M95](#), [M139](#),
 [M267](#), [M278](#), [M395](#), [M443](#),
 [M467](#), [M781](#), [M837](#), [M1814](#),
 [M2172](#), [M2189](#), [I32](#), [I210](#), [I235](#)
`\if@twoside` .. [M95](#), [M138](#), [M609](#), [M667](#)
`\ifcsname` .. [d326](#), [d343](#), [t38](#), [t47](#), [N579](#)
`\ifdt@p` [B190](#), [B192](#)
`\iff` [u492](#)
`\IfFileExists` [88](#), [557](#), [a178](#),
 [k239](#), [k280](#), [k316](#), [k326](#), [k358](#), [Q484](#)
`\IfFileExists@` [k241](#), [k242](#), [k278](#)
`\iffontchar` [I1010](#), [I1158](#)
`\ifG@refundefined` [z3](#), [z4](#), [z5](#)
`\ifh@` [B76](#), [B114](#), [B123](#)
`\ifin@` [I1488](#), [I1491](#), [r50](#), [r52](#),
 [s1](#), [s22](#), [s250](#), [s352](#), [s354](#), [s415](#),
 [s428](#), [s498](#), [s500](#), [s528](#), [s580](#),
 [s594](#), [s628](#), [s640](#), [s689](#), [s705](#),
 [s778](#), [s781](#), [s803](#), [s837](#), [s840](#),
 [s861](#), [s893](#), [s896](#), [s958](#), [s961](#),
 [s964](#), [s1031](#), [s1033](#), [s1062](#), [t157](#),
 [t160](#), [t163](#), [N113](#), [N131](#), [N237](#), [N249](#)
`\ifinner` [B231](#),
 [B239](#), [B259](#), [B286](#), [I57](#), [I126](#), [I315](#)
`\ifmath@fonts` [o168](#), [p168](#), [q176](#)
`\ifmaybe@ic` [w82](#), [w91](#)
`\ifnot@nil` [q297](#), [q314](#), [q335](#)
`\ifodd` [s1005](#), [F172](#),
 [F192](#), [M21](#), [M138](#), [M610](#), [M668](#),
 [M982](#), [M985](#), [M1018](#), [M1021](#),
 [M1132](#), [M1135](#), [M1294](#), [M1297](#),
 [M1574](#), [M1577](#), [M1695](#), [M1698](#),
 [M1818](#), [M2039](#), [M2047](#), [I68](#), [I137](#)
`\IfTargetDateBefore` [N1010](#)
`\iftc@forced` [x784](#), [x795](#), [x1063](#)
`\ifv@` [B75](#), [B113](#), [B122](#)
`\ifvbox` [M319](#), [M376](#), [M423](#), [M502](#), [M518](#)
`\ignorespaces`
 [i49](#), [i117](#), [i136](#), [i148](#), [i159](#), [i175](#),
 [i188](#), [i381](#), [k67](#), [k122](#), [I72](#), [o250](#),
 [p250](#), [A139](#), [A170](#), [A171](#), [B267](#),
 [B294](#), [C55](#), [C217](#), [D109](#), [D334](#),
 [E57](#), [E58](#), [E71](#), [E80](#), [E93](#), [E97](#),
 [E104](#), [E111](#), [E113](#), [E122](#), [E142](#),
 [E213](#), [E277](#), [E279](#), [E281](#), [E308](#),
 [F16](#), [F24](#), [F35](#), [F54](#), [F55](#), [G30](#),
 [G32](#), [H110](#), [K7](#), [K9](#), [I17](#), [I24](#), [I419](#)
`\ignorespacesafterend` [A7](#)
`\IJ` [I250](#), [I433](#), [I547](#), [I1092](#)
`\ij` [I249](#), [I431](#), [I546](#), [I1093](#)
`\Im` [u306](#)
`\imath` [u301](#)
`\in` [u421](#), [u453](#)
`\in@` [I1486](#), [I1489](#), [r49](#), [r51](#),
 [s1](#), [s21](#), [s249](#), [s351](#), [s353](#), [s411](#),
 [s424](#), [s497](#), [s499](#), [s526](#), [s578](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

s589, s626, s637, s687, s701,
 s776, s779, s799, s835, s838,
 s858, s891, s894, s955, s959,
 s962, s1029, s1032, s1060, t156,
 t159, t162, N112, N129, N234, N248
 \in@ s5, s6, s7, s9
 \in@false s10
 \in@true s12
 \in_callback 592, P733
 \include 88, k156, k171, k173
 \IncludeInRelease
 . . . a18, a23, a290, a307, b49,
 b88, b103, b119, b125, b134,
 b139, b148, b154, b168, b182,
 b186, b220, b233, b278, b446,
 b481, b488, b535, c66, d246,
 d274, d300, d306, d322, d352,
 d395, d412, d421, d441, i5, i21,
 i54, i65, i106, i126, i140, i152,
 i165, i180, i221, i257, i339, i345,
 i359, i367, k12, k70, k148, k165,
 k224, k270, k311, k323, k335,
 k346, l77, l104, l148, l169, l324,
 l332, l353, l369, m24, m30, m46,
 m90, m127, m143, m151, m169,
 n5, n11, o174, o196, o355, o366,
 o374, o406, o414, o428, o449,
 o485, o572, o634, p174, p196,
 p355, p366, p374, p406, p414,
 p428, p449, p485, p572, p634,
 q495, q504, r2, r22, s49, s78,
 s138, s169, s575, s623, s669,
 s678, s773, s832, s927, s937,
 t33, t173, t218, t252, t265, u61,
 u79, u98, u104, u614, u626,
 w27, w34, x2, x42, x59, x537,
 z38, z49, A45, A72, A91, A142,
 A155, A161, A176, A192, A216,
 A238, A265, A300, A331, A341,
 B79, B87, B109, B118, B139,
 B144, B152, B163, B226, B234,
 B244, B271, B382, B394, B406,
 B415, C125, C133, D4, D14,
 D72, D80, D136, D144, D190,
 D199, D236, D258, D338, D344,
 D364, D372, E125, E145, F241,
 F287, F422, F438, H5, H20,
 H161, H168, H181, H204, L20,
 L40, M24, M54, M151, M179,
 M345, M366, M371, M419,
 M591, M651, M794, M812,
 M873, M894, M930, M954,
 M1066, M1217, M1386, M1468,
 M1562, M1684, M1903, M1930,
 M2160, M2178, M2197, M2243,
 N60, N73, N103, N122, N152,
 N167, N277, N289, N380, N409,
 N568, N699, P3, P222, P245,
 P268, Q8, Q16, Q23, Q38, Q57,
 Q66, Q73, Q99, Q132, Q219,
 Q224, Q244, Q337, I35, I105,
 I206, I232, I280, I294, I383, I392
 \includeonly 88, k146, k166, k167
 \indent i364, C161, E69
 \index 472, H176,
 J6, J18, L24, L32, M620, M679
 \indexentry J15
 \inf B25
 \infty u308
 \init@restore@glb@settings
 q219, q222, q224
 \init@restore@version
 s62, s91, s108, s123, s124
 \init@series@setup t154, t212, Q544
 \initcatcodetable P90
 \input 88, 557,
 a68, a174, a177, a234, d7, k332,
 q16, r106, t348, t358, t368, u10,
 u11, u12, u13, u14, u22, u39,
 u40, u44, u45, u128, u129, u130,
 u131, u646, u647, u648, x1042,
 N306, P16, Q97, Q111, Q136,
 Q212, Q301, Q489, Q525, Q545
 \input@path
 . . . 1, 6, a109, a131, a133, a139,
 a141, a147, a149, a154, a156,
 a166, a233, k245, k259, k283, k298
 \inputencodingname Q278, Q300, Q382
 \InputIfFileExists
 88, 557, k310, k324,
 k325, k338, k349, k359, k375,
 l1476, o326, p326, t340, t350,
 t360, x758, x1125, N467, O8, Q206
 \inputlineno a327, g165
 \insecunt b37, b51,
 b52, b53, b62, b90, b91, b92,
 b94, b236, b237, b238, b239,
 b240, b241, b252, b253, b254,
 b255, b256, b260, b262, b281,
 b282, b283, b284, b285, b286, M61
 \insert b243, b268, b270, b273,
 b288, M517, M518, M1883, I408
 \install@mathalphabet
 o522, o539, o546,
 p522, p539, p546, s269, s272,
 s358, s359, s456, s508, s511,
 s518, s533, s534, s541, s1045, s1047
 \int u340

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\interdisplaylinepenalty`
 i13, [B55](#), B194, B346
`\interfootlinepenalty` [b349](#)
`\interfootnotelinepenalty`
 b349, i18, I410
`\interlinepenalty`
 i11, o598, p598, A227,
 A230, A248, A251, H67, H118,
 H188, H211, M338, M1153,
 M1157, M1319, M1323, I410
`\intextsep` M1136, M1140, M1155,
 M1158, M1165, M1298, M1304,
 M1321, M1324, M1333, [M2303](#)
`\intop` u339, u340
`\iota` u268
`\is@range` q330, q331
`\ishortstack` [F43](#)
`\itdefault` t30, [u92](#)
`\item` g230, A172, A202, A204, A218,
 A240, B389, B401, B428, [C141](#),
 C219, E66, G36, G38, K4, K8
`\itemindent` . [C9](#), C42, C95, C187, C208
`\itemize` C242
`itemize (environment)` [C242](#)
`\itemsep` [C1](#), C176
`\iterate` a81, a82, [b379](#)
`\itshape`
 I444, I801, t28, t29, t229, t262,
 t268, w21, x554, G36, G38, I375
- J**
- `\J` Q190, Q453
`\j` I248, I403,
 I545, I773, I1101, I1307, I1387, Q462
`\jmath` u302
`\Join` t323
`\joinrel`
 u463, u470, u472, u474, u476,
 u478, u480, u482, u484, u488, u490
`\jot` [B53](#), B191, B353
- K**
- `\k` . I482, I584, I589, I611, I616, I692,
 I693, I751, I752, I806, I808, I813,
 I815, I1193, I1261, I1262, I1279,
 I1280, I1302, I1303, I1304, I1357,
 I1358, I1385, I1386, x194, x212
`\kappa` u269
`\ker` B27
`\kernel@ifnextchar` c76,
 d56, d75, d125, [d369](#), d384, N194
`\kernel@make@fragile`
 d304, d444, d445, d446,
 d447, d448, d449, d450, d451,
 d452, d453, d454, d455, d456,
 d457, d458, d459, i24, i25, i26,
 i27, i28, I172, I173, A144, A145,
 A195, A196, A197, B90, B91,
 B92, B93, B166, B167, B168,
 E148, E149, E150, F441, F442,
 F443, F444, F445, F446, F447,
 F448, F449, F450, F451, F452,
 H23, H24, H25, H26, H27, L43, L44
`\kill` E142, E150
- L**
- `\L` I242, I421, I527, I765,
 I1094, N688, N779, N801, Q481
`\l` .. I251, I423, I548, I774, I1095, Q481
`\l@ngrel@x` .. d49, d50, [d51](#), d95, d142
`\l@nohyphenation` .. A223, A337, [Q214](#)
`\label` [z32](#), H176, L24, L32, M619, M678
`\labelformat` ... [333](#), z39, [z44](#), z50, z56
`\labelsep` .. [C9](#), C210, C216, G36, G38
`\labelwidth` [C9](#), C93, C209, C211, C214
`\Lambda` u292
`\lambda` u270
`\land` u360, u362
`\angle` u586
`\language` b35, b82, b84,
 b99, k50, A223, A337, M597, O10
`\last@fontshape` o455, o473,
 o490, o507, p455, p473, p490, p507
`\lastbox`
 o616, p616, B180, B181, C130,
 C136, C185, H99, H132, M305
`\LastDeclaredEncoding`
 ... o101, [o104](#), p101, [p104](#), Q378
`\lastnamedcs` d344
`\lastnodetype` o609, o610,
 o611, o615, p609, p610, p611, p615
`\lastpenalty` .. o612, p612, w112, w115
`\lastskip` b403, b404, b406,
 b408, i44, i102, i114, i133, i194,
 i195, i199, i201, i202, i210, i230,
 i233, i265, i268, i269, w102,
 w105, C115, C116, C150, C151, F36
`\LaTeX` [j3](#), j15, N651, N741
`\LaTeXe` [j13](#)
`\latexreleaseversion` [c1](#)
`\lbrace` I308, u590
`\lbrack` [b363](#)
`\lccode` g19, g20, g21,
 g22, g23, g24, I140, I1015, A286,
 A312, A355, Q157, Q174, Q182,
 Q189, Q191, Q192, Q194, Q196,
 Q197, Q198, Q199, Q437, Q445,
 Q452, Q454, Q455, Q457, Q459

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \lceil u594
- \ldotp u493, u496, u611
- \ldots l322, u497
- \le u408, u410
- \leaders b435, u332, u550,
u551, u553, u554, E352, F274,
F281, F318, F324, H193, H216
- \leadsto t326
- \leavevmode b394, b421, b424,
b435, b437, i317, i331, l75,
l184, l289, l290, l393, l422, l426,
l429, l476, l757, l790, w123,
x119, x840, A227, A248, A261,
A272, A280, A333, A343, A356,
B389, B401, B428, C58, C103,
D8, D17, D24, D111, D113,
D129, D157, D218, D294, D351,
D368, D375, E166, F45, F167,
H40, H189, H201, H212, K14,
M157, M162, M184, M189, I433
- \leavevmode@ifvmode
... i361, i362, i363, i364, i372,
u616, u618, u620, u622, B115, B141
- \left . u617, u619, u621, u623, u628,
u629, u630, u631, B154, B160, B182
- \Leftarrow u402, u484, u490
- \leftarrow
u429, u431, u472, u482, u488, u542
- \leftarrowfill u526, u542
- \lefteqn B365
- \leftharpoondown u445, u459
- \leftharpoonup u444
- \leftthypenmin O11
- \leftline D402
- \leftmargin
. C9, C52, C53, C94, C146, C148
- \leftmargini B381, C17
- \leftmarginii C17
- \leftmarginiii C17
- \leftmarginiv C17
- \leftmarginv C17
- \leftmarginvi C17
- \leftmark L48
- \Leftrightarrow u401
- \leftrightharrow u428
- \leftskip
b416, o593, p593, A180, A184,
A188, A220, A242, C74, D250,
D271, H186, H191, H209, H214
- \legacyoldstylenums x1, x52
- \leq u406, u408
- \lfloor u598
- \lg B4
- \lggroup u600
- \lhd t329
- \lhook u469, u470
- \lim B6
- \liminf B8
- \limits u531, u535, B149, B307
- \limsup B7
- \line g219, F57, F236, F427, F444
- \linebreak 71, i9, i26
- \linepenalty b307
- \lineskip b357, b389,
b424, u450, B187, D252, D272,
E59, E186, F47, F168, M622, M681
- \lineskiplimit .. b358, b389, b426,
b427, u450, u502, B189, B193,
D238, D253, D260, M622, M681
- \linespread o251, p251
- \linethickness F41, F428, F445
- \linewidth . k28, k86, B252, B278,
B390, B402, B429, B433, B451,
C15, C51, C52, C54, D248,
D269, E36, M146, M205, I266
- \list C34, C236, C247
- \listfiles 557, k387
- \listparindent C9, C41, C50
- \ll u426
- \llap C238, C249, D406, D407
- \lmoustache u555
- \ln B5
- \lnot u318, u319
- \load@onefilewithoptions N825
- \LoadClass 555, N312,
N326, N484, N543, N551, N552
- \LoadClassWithOptions 555, N325
- \LoadFontDefinitionFile
..... o353, o379, o380, p353,
p379, p380, u20, u25, u26, u27, u31
- \loccount P15
- \log B3
- \loggingall b446
- \loggingoutput b442, b455, b471, b485
- \Longleftarrow u484
- \longleftarrow u481
- \Longleftrightharrow u490, u492
- \longleftrightharrow u488
- \longmapsto u486
- \Longrightarrow u478
- \longrightarrow u479, u486
- \loop a81, b379, o607,
p607, E358, N597, N660, N712,
N750, P150, P159, Q261, Q272,
Q282, Q293, Q323, Q349, Q359
- \lor u361, u363
- \lower j2, u450, D169,
F15, F76, F163, F164, F201, F202

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\lower@bound` [q340](#), [q341](#), [q352](#)
`\lowercase` [g26](#), [l141](#),
[l1016](#), [l1474](#), [o267](#), [o325](#), [p267](#),
[p325](#), [A290](#), [A316](#), [A356](#), [Q477](#)
`\lq` [b361](#)
`\lrbox` [D99](#)
`lrbox` (environment) [379](#)
`\ltx@sh@ft` [b430](#),
[l390](#), [l397](#), [l473](#), [l481](#), [l754](#), [l761](#)
`\luabytecode` [P193](#)
`\luachunk` [P201](#)
`\luafunction` [P177](#)
`\luatexbase` [P275](#)
`\luatexluafunction` [a18](#), [a23](#)
`\luatexversion` [a11](#), [P5](#)
- M**
- `\M` [b359](#)
`\m@ne` [b39](#)
`\m@th` [b410](#), [b422](#), [j13](#),
[u329](#), [u451](#), [u453](#), [u454](#), [u457](#),
[u498](#), [u522](#), [u525](#), [u529](#), [u532](#),
[u539](#), [u542](#), [u549](#), [u552](#), [u634](#),
[B68](#), [B71](#), [B106](#), [B136](#), [B154](#),
[B156](#), [B172](#), [B191](#), [B320](#), [B390](#),
[B402](#), [B429](#), [B439](#), [D233](#), [D362](#),
[E169](#), [H193](#), [H216](#), [I376](#), [I381](#), [I389](#)
`\magstep` [b350](#)
`\magstephalf` [b350](#)
`\makeatletter` [d392](#), [k30](#), [k88](#), [o331](#),
[p331](#), [A19](#), [H151](#), [M2](#), [N306](#), [N445](#)
`\makeatother` [d392](#), [N306](#), [Q546](#)
`\makebox` [379](#), [B252](#), [B278](#), [D3](#)
`\makeglossary` [472](#), [k133](#), [J20](#)
`\makeindex` [472](#), [k132](#), [J3](#)
`\makelabel`
[C45](#), [C97](#), [C205](#), [C218](#), [C238](#), [C249](#)
`\MakeLowercase` [Q471](#), [Q483](#)
`\makeph@nt` [B101](#), [B103](#)
`\MakeRobust` [d245](#),
[d423](#), [d424](#), [d425](#), [d426](#), [d427](#),
[d428](#), [d429](#), [d430](#), [d431](#), [d432](#),
[d433](#), [d434](#), [d435](#), [d436](#), [d437](#),
[d438](#), [s673](#), [s932](#), [F424](#), [F425](#),
[F426](#), [F427](#), [F428](#), [F429](#), [F430](#),
[F431](#), [F432](#), [F433](#), [F434](#), [F435](#)
`\makesm@sh` [B131](#), [B133](#)
`\maketitle` [441](#)
`\MakeUppercase` [z46](#), [Q461](#), [Q461](#)
`\mandatory@arg` .. [q368](#), [q455](#), [q459](#),
[q464](#), [q471](#), [q473](#), [q478](#), [q480](#),
[q485](#), [q487](#), [q500](#), [q516](#), [q523](#), [q525](#)
`\mapsto` [u436](#)
`\mapstochar` [u435](#), [u436](#), [u486](#)
- `\marginpar` [I308](#)
`\marginparpush` [M85](#), [M1834](#)
`\marginparsep` ... [M84](#), [M1845](#), [M1847](#)
`\marginparwidth` ... [M83](#), [M1847](#), [I337](#)
`\mark` [L27](#), [L35](#), [L53](#)
`\markboth` [L21](#), [L22](#), [L41](#), [L43](#)
`\markright` [L22](#), [L44](#)
`\marks` [P35](#), [Q10](#), [Q12](#)
`\math` [B299](#)
`math` (environment) [B299](#)
`\math@bgroup`
... [o553](#), [p553](#), [q260](#), [q266](#), [s53](#),
[s81](#), [s142](#), [s172](#), [w130](#), [w131](#), [w138](#)
`\math@egroup` [o553](#),
[p553](#), [q264](#), [q265](#), [w131](#), [w132](#), [w139](#)
`\math@fonts` [o523](#), [o528](#), [p523](#), [p528](#),
[q186](#), [q290](#), [s60](#), [s89](#), [s149](#), [s180](#)
`\math@fontsfalse`
... [j7](#), [l302](#), [l329](#), [l360](#), [l1197](#),
[o41](#), [o170](#), [o180](#), [o203](#), [p41](#),
[p170](#), [p180](#), [p203](#), [x124](#), [x588](#), [x845](#)
`\math@fontstrue` [o168](#), [o565](#), [p168](#), [p565](#)
`\math@version`
... [o7](#), [o271](#), [o527](#), [o531](#), [o533](#),
[o534](#), [o536](#), [p7](#), [p271](#), [p527](#),
[p531](#), [p533](#), [p534](#), [p536](#), [q184](#),
[s56](#), [s59](#), [s64](#), [s65](#), [s69](#), [s84](#), [s88](#),
[s93](#), [s94](#), [s98](#), [s111](#), [s112](#), [s113](#),
[s126](#), [s127](#), [s128](#), [s145](#), [s148](#),
[s152](#), [s154](#), [s156](#), [s160](#), [s175](#),
[s179](#), [s183](#), [s185](#), [s187](#), [s191](#), [t292](#)
`\mathaccent` [s590](#), [s638](#), [s672](#), [s682](#)
`\mathalpha` [s760](#), [s1003](#), [u161](#), [u162](#),
[u163](#), [u164](#), [u165](#), [u166](#), [u167](#),
[u168](#), [u169](#), [u170](#), [u171](#), [u172](#),
[u173](#), [u174](#), [u175](#), [u176](#), [u177](#),
[u178](#), [u179](#), [u180](#), [u181](#), [u182](#),
[u183](#), [u184](#), [u185](#), [u186](#), [u187](#),
[u188](#), [u189](#), [u190](#), [u191](#), [u192](#),
[u193](#), [u194](#), [u195](#), [u196](#), [u197](#),
[u198](#), [u199](#), [u200](#), [u201](#), [u202](#),
[u203](#), [u204](#), [u205](#), [u206](#), [u207](#),
[u208](#), [u209](#), [u210](#), [u211](#), [u212](#),
[u213](#), [u214](#), [u215](#), [u216](#), [u217](#),
[u218](#), [u219](#), [u220](#), [u221](#), [u222](#),
[u289](#), [u290](#), [u291](#), [u292](#), [u293](#),
[u294](#), [u295](#), [u296](#), [u297](#), [u298](#),
[u299](#), [u508](#), [u509](#), [u510](#), [u511](#),
[u512](#), [u513](#), [u514](#), [u515](#), [u517](#), [u520](#)
`\mathbf` [t14](#), [t78](#), [t190](#), [u143](#)
`\mathbin`
... [s1008](#), [u224](#), [u225](#), [u227](#), [u350](#),
[u351](#), [u352](#), [u353](#), [u356](#), [u357](#),
[u358](#), [u359](#), [u362](#), [u363](#), [u364](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

u365, u366, u367, u368, u369,
 u370, u371, u372, u373, u374,
 u375, u376, u377, u378, u379,
 u380, u381, u382, u383, u384,
 u385, u386, u387, u388, u389, B37
 \backslash mathcal u142
 \backslash mathchar
 b422, s701, s745, u327, u328, u609
 \backslash mathchar@type .. s672, s682, s733,
 s736, s745, s761, s930, s941, s1004
 \backslash mathchardef
 b21, b22, b23, b24, b107, b110,
 b111, e3, e4, e5, e6, l70, s736, P214
 \backslash mathcharzero P214
 \backslash mathchoice B61
 \backslash mathclose s1011, u223, u232,
 u234, u237, u242, u248, u250,
 u252, u558, u585, u589, u593,
 u597, u603, B43, B46, B49, B52
 \backslash mathcode s733, u244, u245, u246
 \backslash mathdollar l307, u606
 \backslash mathellipsis l321, u611
 \backslash mathgroup b79, o14,
 p14, q257, q263, q269, q270,
 q281, u635, x8, x14, x49, x1052
 \backslash mathhexbox b422, t310
 \backslash mathindent
 .. B379, B391, B403, B431, B441
 \backslash mathinner ... u496, u500, u505, u611
 \backslash mathit t29, u145, u148, u609
 \backslash mathnormal u141
 \backslash mathop s1007, u333, u334, u335, u336,
 u337, u338, u339, u341, u342,
 u343, u344, u345, u346, u348,
 u349, u529, u532, B3, B4, B5,
 B6, B7, B8, B9, B10, B11, B12,
 B13, B14, B15, B16, B17, B18,
 B19, B20, B21, B22, B23, B24,
 B25, B26, B27, B28, B29, B30,
 B31, B32, B33, B34, B149, B307
 \backslash mathopen s1010,
 u233, u236, u241, u247, u249,
 u251, u556, u587, u591, u595,
 u599, u601, B41, B44, B47, B50
 \backslash mathord s760, s1006, u228,
 u235, u238, u243, u255, u256,
 u257, u259, u260, u261, u262,
 u263, u264, u265, u266, u267,
 u268, u269, u270, u271, u272,
 u273, u274, u275, u276, u277,
 u278, u279, u280, u281, u282,
 u283, u284, u285, u286, u287,
 u288, u300, u301, u302, u303,
 u304, u305, u306, u307, u308,
 u309, u310, u311, u312, u313,
 u314, u315, u316, u317, u319,
 u320, u321, u322, u323, u324,
 u325, u326, u516, u518, u519,
 u541, u542, u545, u546, u547,
 u548, u560, u562, u564, u567,
 u569, u583, u605, u606, u607, u608
 \backslash mathpalette u449,
 u453, u456, B60, B69, B99, B129
 \backslash mathparagraph l310, m134, m146, u606
 \backslash mathph@nt B99, B105
 \backslash mathpunct s1012,
 u226, u230, u493, u494, u495
 \backslash mathrel .. s1009, u229, u231, u239,
 u240, u253, u254, u330, u390,
 u391, u392, u393, u394, u395,
 u396, u397, u398, u399, u400,
 u401, u402, u403, u406, u407,
 u410, u411, u412, u413, u414,
 u415, u416, u417, u418, u419,
 u420, u421, u422, u424, u425,
 u426, u427, u428, u429, u430,
 u433, u434, u435, u437, u438,
 u439, u440, u441, u442, u443,
 u444, u445, u446, u447, u449,
 u453, u456, u463, u465, u468,
 u469, u471, u474, u476, u571,
 u573, u575, u577, u579, u581,
 B42, B45, B48, B51, B149, B307
 \backslash mathring u520
 \backslash mathrm t5, t102, t196, u140
 \backslash mathsection .. l311, m133, m145, u606
 \backslash mathsf t8, t107, t199, u144, u147
 \backslash mathsm@sh B129, B135
 \backslash mathsterling l319, u606
 \backslash mathstrut B84, B93, B158, B159
 \backslash mathsurround b410
 \backslash mathsymbol s738
 \backslash mathtt t11, t112, t202, u146
 \backslash mathunderscore u606
 \backslash mathversion ... o271, p271, t289, t291
 \backslash matrix B156, B160, B167
 \backslash max B22
 \backslash maxdeadcycles M7
 \backslash maxdepth b331,
 i237, k57, k112, M92, M169,
 M170, M506, M514, M546,
 M715, M724, M764, M991, Q85
 \backslash maxdimen b296, b332,
 b333, b389, b427, b443, b454,
 b470, b485, o580, o590, o625,
 o640, p580, p590, p625, p640,
 q338, q391, u450, F247, F292,
 F331, M291, M1853, M1873,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- M1878, M2165, M2205, M2206,
 M2208, N822, N863, N872, Q89
 \maybe@ic w63, w64, w83
 \maybe@ic@ w83
 \maybe@icfalse w97
 \maybe@ictrue w87
 \maybe@load@fontshape l71, t130
 \mb@b D50, D60
 \mb@l D50, D54, D59, F48, F52
 \mb@r D50, D54, D59, F48, F52
 \mb@t D51, D58
 \mbox 379, b422, j13, l293, t306, u498,
 D11, D20, D24, F20, I381, I389
 \mddef@ult t74, t148
 \mddefault t18, t74, t96, t194, u90, u112
 \mdseries
 t16, t17, t89, t192, t193, t309, w20
 \mdseries@rm t60, t93, t183
 \mdseries@sf t60, t94, t184
 \mdseries@tt t60, t95, t185
 \meaning a219,
 a228, a323, d202, d261, d288,
 d390, s412, s425, s526, s591,
 s638, s702, s801, s859, s956, s1060
 \medbreak b404, d430, d451
 \medmskip u637, B36, B38, B202
 \medskip b407, i310
 \medskipamount b406, i311, i313
 \MessageBreak
 d179, d251, d278, g3, g6, g13,
 g33, g46, g60, g73, g171, g173,
 g179, g186, l161, l971, l1479,
 l1482, o471, o505, p471, p505,
 q20, q21, q67, q88, q281, q432,
 q452, q484, q500, q515, q528,
 r31, r33, s367, s376, s514, t55,
 w144, x23, x100, x102, x121,
 x764, x766, x767, x768, x770,
 x772, x773, x774, x775, x776,
 x826, x828, x835, x842, x1057,
 A23, M578, M1960, M1997,
 N145, N336, N347, N349, N351,
 N362, N452, N453, N455, N456,
 N457, N459, N461, N479, N480,
 N481, N482, N528, N545, N546,
 N619, N638, N677, N733, N767,
 N876, N878, N960, N963, N976,
 N978, Q230, Q231, Q232, Q234
 \mho t322
 \mid u394
 \min B23
 \minipage D281
 minipage (environment) 380
 \mit t371
 \mkern u327, u330,
 u332, u454, u463, u505, u506,
 u507, u537, u538, u539, u540,
 u541, u542, u543, u544, B36,
 B37, B40, B73, B74, H194, H217
 \models u476
 \module_error 591, P331
 \module_info 591, P331
 \module_warning 591, P331
 \modules P284
 \month a185, c11, c17, N655, N745
 \moveright M625, M684
 \mp u381
 \mscount E355
 \mskip i336,
 B36, B38, B201, B202, B203, B204
 \mu u271
 \mubyte Q248
 \multicolumn E209
 \multiput F25, F29, F429, F446
 \multispan E209, E355
 \mskip . b29, b55, b93, u537, u538, P32
 \mskipdef b55, b93, P215
 \mskipzero P215

 N
 \n P318, P320, P327,
 P329, P456, P557, P580, P611,
 P628, P650, P658, P659, P679,
 P692, P699, P700, P707, P719
 \n@space u617, u619, u621,
 u623, u628, u629, u630, u631, u634
 \nabla u311
 \narrower b415
 \natural u321
 \ncallback P616
 \ndefault P620
 \ne u405
 \nearrow u397
 \NeedsTeXFormat q12, x727, N341, N1022
 \neg u317, u318
 \negthinspace i361, i370
 \neq u404
 \new@command
 . d52, d53, d106, d140, d159, d213
 \new@environment ... d121, d122, d134
 \new@fontshape r2, r4, r22, r24
 \new@mathalphabet ... s409, s430, s441
 \new@mathgroup b78,
 b80, b98, b100, o14, p14, s289, P25
 \new@mathversion s20, s246, s248
 \new@symbolfont s290, s322
 \new_attribute 589, P397
 \new_bytocode 589, P431

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\new_chunkname` 589, [P444](#)
- `\new_luafunction` 589, [P460](#)
- `\new_whatsit` 589, [P419](#)
- `\newattribute` 588, [P74](#), [P225](#)
- `\newbox` [b47](#), [b301](#), [b412](#),
e13, [A283](#), [B66](#), [C27](#), [D70](#),
[E16](#), [E17](#), [E18](#), [E319](#), [F6](#), [F356](#),
[F361](#), [M86](#), [M120](#), [M121](#), [M122](#)
- `\newcatcodetable` 588,
[P84](#), [P93](#), [P94](#), [P120](#), [P121](#), [P229](#)
- `\newcommand` 37,
[d52](#), [l4](#), [t35](#), [u49](#), [u50](#), [u51](#), [u52](#),
[u54](#), [u55](#), [u57](#), [u58](#), [u90](#), [u91](#),
[u92](#), [u93](#), [u94](#), [u95](#), [u111](#), [u112](#),
[u114](#), [F368](#), [M2291](#), [M2294](#),
[M2297](#), [M2298](#), [M2301](#), [M2302](#)
- `\newcount` [b47](#), [b349](#), [e7](#), [e8](#), [i98](#), [k9](#),
[m36](#), [q25](#), [s27](#), [s254](#), [B55](#), [B311](#),
[B312](#), [C23](#), [C24](#), [C25](#), [C26](#), [C56](#),
[C226](#), [C241](#), [D322](#), [E11](#), [E12](#),
[E13](#), [E14](#), [E15](#), [E311](#), [E312](#),
[E313](#), [F350](#), [F351](#), [F352](#), [F353](#),
[F362](#), [H36](#), [H140](#), [H141](#), [M103](#),
[M105](#), [M107](#), [M109](#), [M111](#),
[M119](#), [M1986](#), [M2289](#), [M2292](#),
[M2295](#), [M2299](#), [N819](#), [Q3](#), [Q4](#),
[Q5](#), [Q77](#), [I3](#), [I267](#), [I268](#), [I269](#), [I270](#)
- `\newcounter` 145, [m10](#)
- `\newdimen` [b47](#),
[b296](#), [b298](#), [b299](#), [b348](#), [e10](#),
[e11](#), [e12](#), [i97](#), [q352](#), [q353](#), [B53](#),
[B380](#), [C9](#), [C10](#), [C11](#), [C12](#), [C13](#),
[C14](#), [C15](#), [C16](#), [C17](#), [C18](#), [C19](#),
[C20](#), [C21](#), [C22](#), [D126](#), [D127](#), [E3](#),
[E5](#), [E6](#), [E7](#), [E8](#), [E154](#), [E314](#),
[E315](#), [E316](#), [E317](#), [F3](#), [F4](#), [F5](#),
[F7](#), [F217](#), [F218](#), [F219](#), [F220](#),
[F221](#), [F222](#), [F354](#), [F355](#), [F357](#),
[F358](#), [F359](#), [F360](#), [M71](#), [M72](#),
[M73](#), [M75](#), [M76](#), [M77](#), [M78](#),
[M79](#), [M80](#), [M81](#), [M82](#), [M83](#),
[M84](#), [M85](#), [M91](#), [M93](#), [M94](#),
[M106](#), [M108](#), [M110](#), [M112](#),
[M113](#), [M114](#), [M115](#), [M116](#),
[M117](#), [M118](#), [M1987](#), [M1988](#), [I398](#)
- `\newenvironment` 37, [d121](#), [N653](#), [N743](#)
- `\newfam` [b47](#), [o16](#), [p16](#), [P36](#)
- `\newfont` [t293](#)
- `\newgroup` [s47](#)
- `\newhelp` [b295](#)
- `\newif` [c68](#), [d143](#), [d417](#), [e9](#), [k7](#),
[k8](#), [o168](#), [p168](#), [s15](#), [w82](#), [x784](#),
[z3](#), [B75](#), [B76](#), [B190](#), [B313](#), [C28](#),
[C29](#), [C30](#), [C31](#), [C32](#), [C33](#), [C138](#),
[D336](#), [E19](#), [E227](#), [F56](#), [F213](#),
[F214](#), [F215](#), [F216](#), [F245](#), [F246](#),
[H38](#), [H124](#), [M95](#), [M96](#), [M97](#),
[M98](#), [M99](#), [M100](#), [M101](#), [M102](#), [N2](#)
- `\newinsert` [b193](#),
[b231](#), [D323](#), [M27](#), [M1852](#), [I366](#)
- `\newlabel` [z22](#), [z34](#)
- `\newlanguage` [b47](#), [Q215](#)
- `\newlength` 152, [n3](#)
- `\newline` [i79](#)
- `\newlinechar` [a72](#), [d5](#)
- `\newluabytecode` 588, [P189](#), [P239](#)
- `\newluachunkname` ... 589, [P197](#), [P241](#)
- `\newluafunction`
..... 588, [P4](#), [P173](#), [P223](#), [P235](#)
- `\newmarks` [Q6](#)
- `\newmathalphabet` [r13](#), [r109](#)
- `\newmathalphabet@` [r14](#)
- `\newmathalphabet@@` [r109](#)
- `\newmathalphabet@@@` [r15](#), [r109](#)
- `\newmuskip` [b47](#)
- `\newpage` [M133](#), [M139](#), [M150](#)
- `\newread` [b47](#), [k3](#)
- `\newsavebox` 379, [D70](#)
- `\newskip` [b47](#), [b297](#),
[b300](#), [b346](#), [b347](#), [e14](#), [e15](#), [e17](#),
[i313](#), [i314](#), [i315](#), [i354](#), [n3](#), [A201](#),
[B314](#), [C2](#), [C3](#), [C4](#), [C5](#), [C6](#),
[C7](#), [C8](#), [M2303](#), [M2304](#), [M2305](#),
[M2309](#), [M2310](#), [M2313](#), [M2314](#),
[M2315](#), [M2319](#), [M2320](#), [M2321](#)
- `\newtheorem` [G1](#)
- `\newtie` [l845](#), [x198](#), [x216](#), [x619](#), [x928](#), [x929](#)
- `\newtoks` [b47](#), [b295](#),
[e16](#), [o281](#), [o282](#), [p281](#), [p282](#), [q201](#)
- `\newwhatsit` 588, [P181](#), [P237](#)
- `\newwrite` [b47](#), [k4](#), [k5](#), [k6](#), [H154](#), [J4](#), [J21](#)
- `\newXeTeXintercharclass` [Q21](#)
- `\nfss@catcodes`
.. [o19](#), [o84](#), [o322](#), [o323](#), [o330](#),
[o377](#), [p19](#), [p84](#), [p322](#), [p323](#),
[p330](#), [p377](#), [u38](#), [u43](#), [u127](#), [M3](#)
- `\nfss@text` [l315](#), [l317](#), [t306](#), [w5](#), [w122](#), [z13](#)
- `\NG` 1528, [l1096](#), [Q481](#)
- `\ng` 1549, [l1097](#), [Q481](#)
- `\ni` [u422](#), [u423](#)
- `\no@alphabet@error`
..... [o4](#), [p4](#), [s268](#), [s270](#),
[s446](#), [s447](#), [s461](#), [s470](#), [s556](#), [s557](#)
- `\noaccents@` [o568](#), [p568](#), [u121](#)
- `\noalign` [u331](#),
[u523](#), [u526](#), [u529](#), [u530](#), [u534](#),
[u535](#), [B158](#), [B159](#), [B175](#), [B178](#),
[B192](#), [B353](#), [E208](#), [E335](#), [E354](#), [F55](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdccl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\nobreak` [b392](#), [b395](#), [b397](#),
[d431](#), [d452](#), [i59](#), [i71](#), [i89](#), [i115](#),
[i121](#), [i134](#), [i147](#), [i173](#), [i297](#), [i305](#),
[i324](#), [i331](#), [i352](#), [k131](#), [k143](#),
[l432](#), [l434](#), [A158](#), [A165](#), [D401](#),
[H90](#), [H191](#), [H192](#), [H196](#), [H214](#),
[H215](#), [H219](#), [L29](#), [L37](#), [M336](#),
[M1149](#), [M1315](#), [Q141](#), [Q143](#),
[Q147](#), [Q148](#), [Q149](#), [Q153](#), [I434](#)
`\nobreakdashes` [i316](#)
`\nobreakspace` [i330](#)
`\nocite` [474](#), [K39](#)
`\nocorr` [w43](#), [w58](#), [w62](#), [w65](#)
`\nocorrlist` [w89](#), [w121](#)
`\nofiles` [88](#), [k127](#)
`\noindent` [o603](#), [o629](#), [p603](#), [p629](#), [H139](#)
`\nointerlineskip`
..... [b387](#), [u331](#), [u523](#), [u526](#),
[u530](#), [u534](#), [B251](#), [B277](#), [F272](#),
[F275](#), [F316](#), [F318](#), [M1842](#), [M1850](#)
`\nolimits` . [u340](#), [u347](#), [B3](#), [B4](#), [B5](#),
[B9](#), [B10](#), [B11](#), [B12](#), [B13](#), [B14](#),
[B15](#), [B16](#), [B17](#), [B18](#), [B19](#), [B20](#),
[B21](#), [B26](#), [B27](#), [B28](#), [B29](#), [B31](#), [B34](#)
`\nolinebreak` [71](#), [i9](#), [i27](#)
`\non@alpherr` [o547](#), [o549](#), [p547](#), [p549](#),
[s72](#), [s101](#), [s117](#), [s163](#), [s194](#), [s1082](#)
`\nonfrenchspacing`
..... [b353](#), [b541](#), [d432](#), [d453](#), [k46](#), [k104](#)
`\nonscript` [B36](#), [B38](#)
`\nonumber` [B340](#), [B363](#), [B364](#)
`\nopagebreak` [71](#), [i7](#), [i25](#)
`\noprotrusion` [H201](#), [H224](#)
`\normalbaselines` .. [b357](#), [B154](#), [B156](#)
`\normalbaselineskip`
..... [b346](#), [b358](#), [q142](#), [D254](#), [D273](#)
`\normalcolor` [B306](#), [B376](#),
[D63](#), [D314](#), [H197](#), [H220](#), [M216](#),
[M492](#), [M629](#), [M639](#), [M688](#),
[M698](#), [M2226](#), [M2259](#), [I97](#), [I166](#)
`\normalfont` [o581](#), [o641](#),
[p581](#), [p641](#), [t311](#), [w18](#), [A262](#),
[B306](#), [B376](#), [H197](#), [H220](#), [I377](#)
`\normallineskip` [b346](#), [b357](#), [D252](#), [D272](#)
`\normallineskiplimit` [b346](#),
[b358](#), [B193](#), [D237](#), [D253](#), [D259](#)
`\normalmarginpar` [I363](#)
`\normalsfcodes` [k42](#), [k44](#), [k46](#), [k100](#),
[k102](#), [k104](#), [k126](#), [M618](#), [M677](#)
`\normalsize` [k40](#), [k98](#), [w142](#),
[M617](#), [M676](#), [N5](#), [I23](#), [I176](#), [I348](#)
`\not` [u330](#), [u404](#), [u405](#), [u427](#)
- `\not@base` [t318](#),
[t322](#), [t323](#), [t324](#), [t325](#), [t326](#),
[t327](#), [t328](#), [t329](#), [t330](#), [t331](#), [t332](#)
`\not@math@alphabet` [t5](#),
[t8](#), [t11](#), [t14](#), [t17](#), [t20](#), [t23](#), [t26](#),
[t29](#), [t78](#), [t90](#), [t102](#), [t107](#), [t112](#),
[t190](#), [t193](#), [t196](#), [t199](#), [t202](#), [t272](#)
`\notin` [u453](#)
`\nu` [u272](#)
`\null` [b371](#), [l327](#), [l363](#),
[l483](#), [l486](#), [l828](#), [l831](#), [l1195](#), [z17](#),
[A227](#), [A248](#), [A333](#), [A343](#), [B112](#),
[B121](#), [B156](#), [B185](#), [H191](#), [H214](#)
`\nulldelimiterspace` [b335](#), [u634](#)
`\nullfont` [A88](#)
`\number` [a86](#), [d2](#), [d89](#),
[m108](#), [o531](#), [o534](#), [p531](#), [p534](#),
[q393](#), [s64](#), [s93](#), [s113](#), [s128](#), [s153](#),
[s184](#), [t303](#), [N559](#), [N655](#), [N745](#), [P105](#)
`\numberline` [H72](#), [H82](#), [H227](#), [I17](#)
`\numexpr` .. [b189](#), [b205](#), [b215](#), [b235](#),
[d333](#), [l994](#), [M36](#), [P82](#), [P105](#), [P157](#)
`\nunknown` [P633](#)
`\narrow` [u399](#)
- ## O
- `\O` .. [l244](#), [l400](#), [l530](#), [l764](#), [l1081](#), [Q480](#)
`\o` .. [l253](#), [l405](#), [l551](#), [l775](#), [l1087](#), [Q480](#)
`\o@lign` [b424](#),
[l390](#), [l397](#), [l473](#), [l481](#), [l754](#), [l761](#)
`\oalign` [b424](#)
`\obeycr` [i378](#)
`\obeylines` [b374](#), [d433](#), [d454](#),
[A233](#), [A254](#), [A324](#), [A325](#), [M583](#)
`\obeyspaces` ... [b374](#), [d434](#), [d455](#), [M583](#)
`\oddsidemargin` [M72](#), [M74](#), [M611](#), [M670](#)
`\odot` [u376](#)
`\OE` .. [l243](#), [l399](#), [l529](#), [l763](#), [l1098](#), [Q480](#)
`\oe` .. [l252](#), [l404](#), [l550](#), [l776](#), [l1099](#), [Q480](#)
`\of` [B67](#), [B310](#)
`\offinterlineskip` [b387](#)
`\oint` [u347](#)
`\ointop` [u346](#), [u347](#)
`\oldstylenums` [x1](#), [x43](#), [x45](#),
[x307](#), [x308](#), [x309](#), [x310](#), [x311](#),
[x312](#), [x313](#), [x314](#), [x315](#), [x316](#), [x1049](#)
`\Omega` [u299](#)
`\omega` [u282](#)
`\ominus` [u379](#)
`\omit` [B178](#),
[B179](#), [E345](#), [E348](#), [E355](#), [E359](#)
`\on@line` [g8](#),
[g15](#), [g165](#), [A97](#), [A114](#), [D105](#), [N478](#)
`\onecolumn` [M141](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\OnlyDescription` q5, v3
`\ooalign` b424,
 l327, l357, l394, l477, l483,
 l485, l496, l512, l725, l758, l828,
 l831, l877, l1195, t308, u454, u457
`\openup` B186, B191
`\operator@font` u635, B3, B4,
 B5, B6, B7, B8, B9, B10, B11,
 B12, B13, B14, B15, B16, B17,
 B18, B19, B20, B21, B22, B23,
 B24, B25, B26, B27, B28, B29,
 B30, B31, B32, B33, B34, B37, B40
`\oplus` u380
`\optional@arg`
 q369, q448, q450, q522, q525
`\OptionNotUsed` N218, N225, N493
`\oslash` u377
`\OT` l372
`\otimes` u378
`\outer` P19, P36
`\outer@nobreak` .. l181, l251, l255, l342
`\outerparskip` C1
`\output` M256
`\outputpenalty` M258, M272,
 M295, M298, M299, M334,
 M1159, M1160, M1325, M1328
`\oval` F236, F239, F430, F447
`\over` u461, B149, B308
`\overbrace` u528
`\overfullrule` b330, L69
`\overleftarrow` u525
`\overrightarrow` u522
`\owns` u423, u424
- P**
- `\P` l310
`\p@` b298
`\p@equation` B318, B438
`\p@reset@font` t311
`\p@selectfont` q117
`\PackageError` c72, c101, c110,
 g84, l1477, x740, x741, x790, x834
`\PackageInfo` g84,
 x759, x764, x780, x781, x841, x1126
`\PackageWarning` g84, x742, x791, x1055
`\PackageWarningNoLine` g84, l969, M1959
`\pagebreak` 71, i6, i7, i22, i24
`\pagegoal` M1880, M1887
`\pagenumbering` 331, y5
`\pageref` z10
`\pageshrink` M538, M542, M558
`\pagestyle` L2
`\pagetotal` M128
`\paperheight` M93
`\paperwidth` M93
`\par` a120,
 b11, b367, b375, b376, b391,
 b400, b401, b402, b404, b406,
 b408, d6, h3, h4, h5, o606, p606,
 A86, A158, A165, A225, A246,
 C63, C110, C127, C129, C135,
 C161, C164, D243, D264, D310,
 E183, E361, H41, H90, H199,
 H221, L62, L63, M166, M193,
 M257, M1886, P156, I15, I24, I249
`\par@deathcycles` C56, C77, C79, C80
`\paragraphmark` H143
`\parallel` u393
`\parbox` 379, D189
`\parboxrestore` D277
`\parfillskip` b345,
 o580, o595, o640, p580, p595,
 p640, A181, A189, A221, A243,
 C76, D251, D272, H186, H209
`\parindent` b337, b416, b417,
 A181, A185, A189, A221, A243,
 C50, D246, D267, H187, H210
`\parsep` C1, C49, C90
`\parseunicodedataI` P123, P162
`\parseunicodedataII` P124, P126
`\parseunicodedataIII` P128, P134
`\parseunicodedataIV` P130, P142
`\parseunicodedataV` P146, P149
`\parshape` C54
`\parskip`
 b338, A169, A219, A221, A241,
 A243, B447, C49, C73, C88,
 C90, C117, C153, C172, C223,
 D246, D267, E67, M1159, M1327
`\partial` u307
`\partopsep` B445, C1, C61
`\PassOptionsToClass` 555, N195
`\PassOptionsToPackage` 555, N195
`\patch@level` c1, c36, c41, c43,
 c45, c48, c56, Q491, Q503, Q505
`\patterns` L205
`\penalty` b396, b397, b398,
 b399, b400, b401, b405, b407,
 b409, i34, i37, i46, i231, i241,
 i266, i270, w118, A227, A230,
 A248, A251, B37, B194, B353,
 C190, E56, K17, M136, M176,
 M195, M198, M1157, M1323,
 I195, I199, I201, I217, I221, I223
`\perp` u439
`\ph@nt` B81, B82, B83, B97
`\phantom` B75
`\Phi` u297

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdccl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \phi u279
- \Pi u294
- \pi u274
- \pickup@font l181, o159,
o288, o482, o516, p159, p288,
p482, p516, q122, q285, q287, q289
- \pictur@ F8
- \picture F8
- \pkgcls@arg N836, N959
- \pkgcls@candidate .. N823, N838,
N914, N918, N922, N990, N993
- \pkgcls@debug .. N813, N829, N830,
N831, N832, N833, N890, N891,
N892, N893, N902, N907, N925,
N934, N949, N983, N984, N985
- \pkgcls@innerdate
N818, N863, N866, N872, N1011
- \pkgcls@mindate
N843, N852, N868, N873
- \pkgcls@name N835, N878
- \pkgcls@parse@date@arg .. N837, N848
- \pkgcls@parse@date@arg@ .. N854, N857
- \pkgcls@parse@date@arg@version .
..... N864, N885
- \pkgcls@releasedate
..... N823, N919, N923, N994
- \pkgcls@rollbackdate@error
..... N915, N973, N991
- \pkgcls@show@selection
..... N942, N947, N997, N1002
- \pkgcls@targetdate
..... N818, N850, N858,
N861, N862, N866, N874, N875,
N888, N896, N911, N913, N943,
N954, N956, N981, N987, N989
- \pkgcls@targetlabel
... N818, N851, N871, N886,
N898, N930, N963, N1001, N1004
- \pkgcls@use@this@release .. N899,
N916, N918, N931, N941, N993
- \pm u382
- \pmatrix B160, B168
- \pmod B39
- \poptabs g206, E130, E149
- \poptracing q130, q294
- \postdisplaypenalty
..... i12, B388, B400, B426
- \pounds l318
- \Pr B32
- \pr@@s B213, B221
- \pr@@t B216, B222
- \pr@m@s B210, B211
- \prec u413
- \preceq u416
- \predisdisplaypenalty
..... b316, B387, B399, B425
- \preload@sizes r11, r94
- \prepare@family@series@update ..
..... t103, t108, t113, t119, t209
- \pretolerance b303,
o582, o597, o642, p582, p597, p642
- \prevdepth . b387, b391, b392, i237,
i238, i295, i300, B192, M167,
M169, M172, I196, I198, I218, I220
- \prim@s B207, B209, B221
- \prime u245, u309, B210
- \prime@s B208
- \process@table k38, k96, s200
- \ProcessOptions I1497,
q71, x757, x794, N226, N269, N547
- \ProcessOptions* N226
- \prod u341
- \propto u390
- \protect d77, d194, d208, d217, d222,
d225, d226, d228, d229, d234,
d235, d240, d243, d244, d266,
d293, g197, g199, g200, g206,
g212, g219, g227, g230, g236,
k139, l26, l32, l51, l55, l209,
l217, s475, s1087, t296, w143,
z12, A103, A106, A121, A131,
E240, H12, H72, H82, H164,
H171, K5, M596, M655, Q254, I17
- \protected
. i56, i364, m160, t248, A47, A157
- \protected@edef d227,
m158, t221, z41, z45, z53, D330,
H60, Q468, Q477, Q482, Q483, I414
- \protected@file@percent
. A43, A50, A65, A73, A74, H165
- \protected@wlog ... N160, N162, N177
- \protected@write
.. k130, k135, z33, H175, J14, J31
- \protected@xdef
d227, H11, N155, I400, I424, I440
- \provide@command d153, d154
- \providecommand
d153, l6, l964, x728, x729, M1970
- \provides_module 591, P285
- \ProvidesClass 555, N182
- \ProvidesFile
a89, u657, u659, u660, u661, N184
- \ProvidesPackage 555, q13,
x734, x762, N141, N182, N1023
- \ProvideTextCommand l3, l60
- \ProvideTextCommandDefault l57
- \ps@empty L10, Q91
- \ps@plain L13

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\Psi` u298
`\psi` u281
`\pushtabs` g206, E126, E127, E146, E148
`\pushtracing` q115, q275
`\put` F21,
 F177, F178, F179, F180, F185,
 F187, F199, F200, F201, F202,
 F207, F210, F405, F431, F448
- Q**
- `\q@curr@file` N608, N610, N615
`\qbezier` 414, F368, F432, F449
`\qbeziermax` F367, F389
`\qquad` i375
`\quad` ... i375, B155, B157, B177, H111
`\quote@name` k236, k274
`\quote@name` k236, k260, k262, k273, N608
`\quotedblbase` l552, l777, l1118
`\quotesinglbase` l553, l1115
- R**
- `\r` b365, b366, l236,
 l388, l428, l467, l604, l631, l641,
 l667, l750, l789, l1187, l1205,
 l1231, l1353, l1354, x195, x213
`\r@t` B66
`\radical` s953, s956, s986
`\raggedbottom` L53
`\raggedleft` A186, A197, A204
`\raggedright` A182, A196, A202
`\raise` l327, l359, l427, l430,
 l726, l791, l878, l1195, t309,
 u457, u505, u507, B73, D386,
 D395, F22, F32, F75, F163,
 F238, F266, F310, F338, F416
`\raisebox` 380, l1164, D363
`\rangle` u584
`\rbrace` l309, u588
`\rbrack` b363
`\rceil` u592
`\Re` u305
`\Ref` 333, z39, z45, z50, z57
`\ref` z10, z45
`\refstepcounter`
 ... 145, z39, z40, z50, z52,
 B304, B427, C202, G27, H59, I9
`\registernumber` 590, P376
`\reinstall@nfss@defs` Q544
`\relbar` u468, u476, u478, u484
`\relbar` u465, u480, u482
`\relpenalty` b311
`\rem@pt` o264, p264
`\remove@angles` q301, q324
`\remove@nil` s36
`\remove@star` q301, q307
`\remove@tlig` 1983,
 1985, 1987, 1994, l1030, l1032, l1034
`\remove@to@nnil`
 ... o263, p263, q301, q327, q440
`\remove_from_callback` 591, P690
`\removelastskip` b403, b405, b407, b409
`\renew@command` . d99, d100, d160, d168
`\renew@environment` d127, d128
`\renewcommand` 37, d99, u64,
 u66, u68, u69, u71, u73, u75,
 u76, u82, u84, u86, u87, u100,
 u101, u107, u108, B375, B395, B416
`\renewenvironment` 37, d127, B424, B436
`\repeat` a81, a83, b379, o623,
 p623, E358, N601, N664, N716,
 N754, P154, P164, Q265, Q276,
 Q286, Q297, Q327, Q353, Q363
`\requestedLaTeXdate`
 ... N817, N850, N870, N956
`\requestedpatchdate` N880, N957
`\RequirePackage` 555, M1967,
 N302, N309, N330, N543, P22
`\RequirePackageWithOptions` 555, N328
`\reserved@a` a121, a125, a126, a195,
 a196, a199, a217, a221, a243,
 a250, a253, a255, a256, a263,
 a266, a268, a269, a276, a279,
 a281, a331, a332, a333, b193,
 c13, c19, c34, d92, d95, d108,
 d109, d110, d112, d159, d160,
 d161, d167, d168, d169, d170,
 d173, d192, d200, d204, d259,
 d263, d286, d290, d366, d375,
 f33, f37, g185, i319, i322, k140,
 k141, k187, k188, k246, k248,
 k253, k255, k257, k263, k267,
 k284, k286, k291, k293, k296,
 k302, k306, k340, k343, k351,
 k354, k369, k370, k374, k380,
 k399, k403, k407, l81, l82, l86,
 l89, l97, l107, l110, l119, l138,
 l143, l977, l981, l1008, l1017,
 o29, o32, o35, o69, o72, o74,
 o111, o115, o324, o327, o454,
 o455, o470, o473, o478, o489,
 o490, o503, o507, o512, o539,
 o542, o543, o551, p29, p32,
 p35, p69, p72, p74, p111, p115,
 p324, p327, p454, p455, p470,
 p473, p478, p489, p490, p503,
 p507, p512, p539, p542, p543,
 p551, q150, q152, q154, q164,
 q166, q169, q298, q299, q312,

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- q313, r53, r57, s356, s365, s367,
s411, s414, s424, s427, s525,
s527, s589, s593, s637, s639,
s700, s703, s799, s802, s858,
s860, s955, s957, s1059, s1061,
s1077, s1079, s1080, s1085, w47,
w48, w53, w54, w65, w68,
w88, w95, A41, A42, A95,
A96, A100, A112, A113, A117,
A149, A150, B355, B356, B357,
B358, B360, D52, D53, D56,
D100, D106, E217, E221, E226,
E245, E336, E337, F79, F81,
F85, F250, F294, F295, M37,
M46, M48, M50, M877, M897,
M1963, M1965, M1966, M2055,
M2057, M2063, M2066, N107,
N115, N119, N125, N133, N137,
N282, N286, N292, N296, N342,
N343, N346, N387, N391, N403,
N404, N406, N415, N419, N431,
N432, N433, N435, N446, N487,
N624, N625, N627, N631, N840,
N845, N897, N898, N929, N930,
N1000, N1001, N1025, N1027,
Q159, Q176, Q177, Q178, Q185,
Q186, Q187, Q387, Q390, Q421,
Q427, Q428, Q439, Q440, Q441,
Q448, Q449, Q450, Q463, Q464,
Q468, Q469, Q472, Q473, Q477,
Q478, Q504, Q507, Q508, Q526,
I29, I30, I32, I33, I63, I67,
I72, I74, I76, I78, I83, I84,
I132, I136, I142, I145, I148, I151
- \reserved@b a122,
a123, d84, d86, d93, d110, d111,
d201, d202, d204, d260, d261,
d263, d287, d288, d290, d367,
d377, f33, f34, f37, i320, i321,
i328, k186, k188, k258, k260,
k262, k297, k299, k301, k402,
k408, l90, l97, l112, l119, l980,
l981, l1017, l1026, l1028, o59,
o61, o114, o115, o540, o551,
p59, p61, p114, p115, p540,
p551, r47, r54, r71, r73, s282,
s284, s337, s339, s364, s365,
s366, s401, s403, s482, s484,
s529, s530, s531, s538, s698,
s702, s704, w52, w53, w66, w68,
w95, w96, E222, E224, E226,
M786, M789, M803, M806,
M823, M826, N108, N109,
N110, N112, N126, N127, N130,
N395, N401, N404, N423, N429,
N433, N626, N627, N630, N631,
N666, N667, N669, N695, N756,
N757, N759, N786, Q162, Q164,
Q168, Q393, Q395, Q399, Q464,
Q473, Q526, I43, I44, I112, I113
- \reserved@c a123, a128,
d372, d375, d377, d380, k391,
k392, o60, o61, o541, o544,
p60, p61, p541, p544, r48, r55,
r61, r68, s33, s37, s283, s284,
s338, s339, s402, s403, s483,
s484, s506, s515, s530, s544,
s789, s808, s817, s848, s865,
s874, s904, s915, s972, s985,
s987, w67, w69, w76, N609,
N614, N615, N635, N643, N649,
N671, N678, N727, N728, N729,
N739, N761, N768, N796, Q166,
Q171, Q179, Q387, Q397, Q418,
Q419, Q420, Q422, Q423, Q424,
Q425, Q426, Q434, Q442, Q528
- \reserved@d
a126, a129, d365, d374, k390,
k392, r61, r68, r70, r74, s797,
s808, s817, s856, s865, s874,
s912, s915, s980, s985, s989, Q529
- \reserved@e
i57, i59, i69, i71, i83, i89, r39,
r45, r70, r73, r74, s34, s39, Q530
- \reserved@f i58,
i59, i70, i71, i89, l1473, l1474,
l1475, l1476, l1478, l1485, o154,
o156, o162, o163, p154, p156,
p162, p163, q336, q347, q351,
q355, q361, q364, q403, q440,
q443, r27, r38, r45, r71, r73, Q531
- \reset@font .. t311, z13, D327, K20,
L14, M616, M675, I175, I347, I409
- \restglb@settings q222, q232
- \restore@mathversion
..... s107, s110, s125, s133
- \restore@protect d227
- \restorecr i378
- \reversemarginpar I363
- \rfloor u596
- \rgroup u600
- \rhd t331
- \rho u275
- \rho hook u471, u472
- \right u617, u619, u621, u623, u628,
u629, u630, u631, B155, B160, B184
- \Rightarrow u403, u478, u490

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\rightarrow` [u430](#),
[u432](#), [u436](#), [u470](#), [u480](#), [u488](#), [u541](#)
 - `\rightarrowfill` [u523](#), [u539](#)
 - `\rightarrowdown` [u447](#)
 - `\rightarrowup` [u446](#), [u458](#)
 - `\righthypenmin` [O11](#)
 - `\rightleftharpoons` [u456](#)
 - `\rightline` [D402](#)
 - `\rightmargin` [C9](#), [C40](#), [C51](#)
 - `\rightmark` [L48](#)
 - `\rightskip` [b417](#), [o594](#), [p594](#),
[A180](#), [A183](#), [A188](#), [A220](#), [A242](#),
[C75](#), [D250](#), [D271](#), [H186](#), [H209](#)
 - `\rlap` [l427](#),
[l430](#), [l791](#), [B365](#), [B376](#), [D406](#), [E69](#)
 - `\rlh@` [u456](#), [u457](#)
 - `\rmdef@ult` [t70](#), [t81](#), [t93](#), [t165](#), [x27](#), [x105](#)
 - `\rmdefault` [t6](#), [t70](#),
[t103](#), [t156](#), [t197](#), [u48](#), [u111](#), [x7](#), [x48](#)
 - `\rmfamily`
[t4](#), [t5](#), [t101](#), [t165](#), [t195](#), [t196](#), [w15](#)
 - `\rmoustache` [u557](#)
 - `\rmsubstdefault` [u17](#), [u28](#), [x28](#), [x39](#), [x106](#)
 - `\Roman` [l45](#), [m104](#)
 - `\roman` [l45](#), [m103](#)
 - `\romannumeral` [m109](#),
[m110](#), [A120](#), [C43](#), [C234](#), [C245](#)
 - `\root` [B66](#), [B310](#)
 - `\rootbox` [B66](#)
 - `\rq` [b361](#)
 - `\rule` [380](#), [D334](#), [D337](#), [I419](#)
- S**
- `\S` [l311](#)
 - `\s@fct@` [q380](#), [q444](#)
 - `\s@fct@alias` [q506](#)
 - `\s@fct@fixed` [q519](#)
 - `\s@fct@gen` [q456](#)
 - `\s@fct@genb` [q461](#)
 - `\s@fct@sgen` [q456](#)
 - `\s@fct@sgenb` [q461](#)
 - `\s@fct@sub` [q468](#)
 - `\s@fct@subf` [q511](#)
 - `\samepage` [71](#), [i11](#), [i28](#)
 - `\savebox` [379](#), [D71](#)
 - `\savecatcodetable` . [P117](#), [P168](#), [P170](#)
 - `\saved@space@catcode` . . . [Q251](#), [Q320](#)
 - `\sb` [B199](#)
 - `\sbox` [379](#), [b411](#), [j4](#), [l495](#), [l511](#),
[C205](#), [D77](#), [D84](#), [D88](#), [D93](#), [D98](#)
 - `\scan@fontshape` [r7](#), [r40](#), [r43](#)
 - `\scan@fontshape` [r6](#), [r26](#), [r37](#)
 - `\scdefault` [t27](#), [u92](#)
 - `\scriptfont` [q292](#)
 - `\scriptfont@name` [q287](#), [q292](#)
 - `\scriptscriptfont` [q293](#)
 - `\scriptscriptstyle` [B65](#), [B68](#)
 - `\scriptspace` [b336](#)
 - `\scriptstyle` [u329](#), [B64](#)
 - `\scshape` [l300](#), [t25](#), [t26](#), [w23](#)
 - `\searrow` [u398](#)
 - `\sec` [B20](#)
 - `\secdef` [H142](#)
 - `\sectionmark` [H143](#)
 - `\select@group` [o524](#),
[o543](#), [p524](#), [p543](#), [s48](#), [s236](#),
[s273](#), [s411](#), [s464](#), [s473](#), [s511](#), [s543](#)
 - `\selectfont` [j7](#), [l302](#), [l329](#),
[l360](#), [l446](#), [l803](#), [l1163](#), [l1197](#),
[l1498](#), [o249](#), [p249](#), [q112](#), [t6](#), [t9](#),
[t12](#), [t15](#), [t18](#), [t21](#), [t24](#), [t27](#),
[t30](#), [t87](#), [t99](#), [t105](#), [t110](#), [t115](#),
[t191](#), [t194](#), [t197](#), [t200](#), [t203](#),
[t239](#), [t299](#), [x36](#), [x115](#), [x125](#),
[x556](#), [x588](#), [x829](#), [x846](#), [l379](#), [l387](#)
 - `\seriesdefault` [l1499](#), [s239](#), [t314](#), [u111](#)
 - `\set@mathdelimiter` [s913](#), [s946](#)
 - `\set@color` [D62](#)
 - `\set@curr@file` [k152](#),
[k160](#), [k222](#), [k240](#), [k271](#), [k276](#), [N607](#)
 - `\set@display@protect`
[d3](#), [d225](#), [g7](#), [g14](#), [g34](#), [g61](#), [N163](#)
 - `\set@fontsize`
[o252](#), [o254](#), [p252](#), [p254](#), [q119](#), [q132](#)
 - `\set@mathaccent`
..... [s598](#), [s606](#), [s641](#), [s649](#), [s667](#)
 - `\set@mathchar` [s722](#), [s732](#)
 - `\set@mathdelimiter`
..... [s805](#), [s814](#), [s862](#), [s871](#), [s925](#)
 - `\set@mathradical` [s244](#), [s982](#)
 - `\set@mathsymbol` [s706](#), [s714](#), [s735](#)
 - `\set@simple@size@args`
..... [q302](#), [q315](#), [q322](#), [q343](#), [q357](#)
 - `\set@size@funct@args` [q305](#), [q307](#), [q365](#)
 - `\set@size@funct@args@` [q365](#)
 - `\set@typeset@protect`
..... [d225](#), [d244](#), [E185](#),
[E211](#), [M603](#), [M605](#), [M661](#), [M663](#)
 - `\setattribute` [589](#), [P82](#), [P226](#)
 - `\setcounter`
.. [l45](#), [k215](#), [m2](#), [m37](#), [C225](#),
[M2290](#), [M2293](#), [M2296](#), [M2300](#)
 - `\setlength`
.. [l52](#), [n4](#), [B443](#), [B448](#), [B449](#),
[B450](#), [D43](#), [D159](#), [D220](#), [D223](#),
[D296](#), [D353](#), [D354](#), [D355](#), [D384](#),
[D385](#), [D392](#), [D393](#), [D394](#), [E164](#),

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- E360, M2306, M2307, M2308,
 M2311, M2312, M2316, M2317,
 M2318, M2322, M2323, M2324
 \SetMathAlphabet o11,
 p11, r140, r141, s480, u147, u148
 \SetMathAlphabet@ ... s418, s487, s496
 \setminusminus u385
 \settrangeatcode P96, P104, P113, P114
 \SetSymbolFont . s335, u137, u138, u139
 \SetSymbolFont@ s308, s342, s350
 \settodepth 152, n17
 \settoheight 152, n17
 \settowidth 152, n17
 \sfsize j6, l302,
 o188, o207, o563, p188, p207,
 p563, q282, q286, x588, I381, I389
 \sfcode ... b353, b354, b355, b356,
 b440, i326, k43, k101, Q183, Q446
 \sfdef@ult t71, t82, t94, t166, x29, x107
 \sfdefault t9, t71, t108, t159, t200, u48
 \sffamily
 t7, t8, t106, t166, t198, t199, w16
 \sfsubstdefault .. u18, u29, x30, x108
 \sh@ft b428
 \shapedefault . l1499, s240, t315, u111
 \sharp u322
 \shipout M602, M660
 \shortstack F43, F433, F450
 \showboxbreadth
 ... b325, b443, b496, b513, b529
 \showboxdepth
 b326, b443, b495, b512, b530,
 o582, o626, o643, p582, p626, p643
 \showhyphens o571, p571
 \showoutput b442
 \showoverfull . b441, b444, b478, b486
 \Sigma u295
 \sigma u276
 \sim u437, u449
 \simeq u438
 \sin B9
 \sinh B11
 \sixt@n a71,
 b16, b64, b66, b96, b97, b98,
 o14, p14, s84, s175, s584, s586,
 s632, s634, s693, s695, s741,
 s743, s785, s787, s793, s795,
 s844, s846, s852, s854, s900,
 s902, s908, s910, s968, s970,
 s976, s978, F136, F151, F153,
 M1005, M1051, M1190, M1358,
 M1592, M1656, M1713, M1783,
 M2009, M2018, M2074, M2090,
 M2123, P28, I62, I80, I131, I153
 \size@update .. q128, q139, q158, q160
 \sizefn@info
 q306, q308, q316, q344, q358
 \skew u536
 \skip ... b28, b53, b92, b208, b239,
 b284, D313, M316, M490, P31, I367
 \skip@ b41,
 b390, b392, b393, b395, w105, w108
 \skipdef b45, b53, b92, P216
 \skipzero P216
 \slash b396, d435, d456
 \sldefault t24, u92
 \sloppy D255, D274, L57, L62
 \sloppypar L62
 sloppypar (environment) L62
 \slshape l437, l794, t22, t23, w22, x547
 \smallbreak b404, d436, d457
 \smallint u349
 \smallskip b405, i310
 \smallskipamount b404, i310, i313
 \smash u465, u539, u540, u543, u544, B126
 \smile u442
 \sp B199
 \sp@n E355
 \space b369
 \spacefactor b394, b395, i103,
 i112, i131, i145, i157, i171, i185,
 i326, i342, i347, l70, l73, I434, I436
 \spaceskip x6, x47
 \spadesuit u326
 \span E359
 \split@name o292, o304, o418, o433,
 p292, p304, p418, p433, q473, q487
 \splitfirstmark M2211
 \splitmaxdepth b332, M2205, I412
 \splittopskip b344, I411
 \sqcap u368
 \sqcup u369
 \sqrt B309
 \sqrtsign u521, B71, B309
 \sqsubset t327
 \sqsubsest u391
 \sqsupset t328
 \sqsupseteq u392
 \SS l304, l531, I1108, Q481
 \ss . l254, l406, l554, l778, I1083, Q481
 \sscshape w31
 \ssf@size
 . l328, l360, l1196, o189, o208,
 o564, p189, p208, p564, q282, q288
 \stackrelrel B307
 \star u389

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\stepcounter` . 145, [m17](#), m27, o536,
 p536, [s48](#), z40, z52, B317, B360,
 B437, M646, M705, I399, I423
`\stop` [A86](#)
`\storedpar` P156, P161
`\stretch` [i356](#)
`\strip@prefix` ... [a111](#), a228, a323,
 d202, d261, d288, d389, [o521](#), [p521](#)
`\strip@pt` b432,
 o180, o186, o187, o188, o189,
 o202, o206, [o264](#), o563, o564,
 p180, p186, p187, p188, p189,
 p202, p206, [p264](#), p563, p564, q134
`\strut` [b412](#), d437, d458, B178, B179, E29
`\strutbox` [b412](#),
 q143, D334, E174, E175, I412, I419
`\sub@sfcnt` ... q468, q469, q470, q497
`\subf@sfcnt` q511, q512, q513
`\subparagraphmark` [H143](#)
`\subsectionmark` [H143](#)
`\subset` u418
`\subseteq` u420
`\subst@correction` . o49, o55, p49, p55
`\subst@fontshape` r8, [r80](#)
`\subst@size` [q419](#)
`\substdefault` x112
`\subsubsectionmark` [H143](#)
`\succ` u412
`\succeq` u415
`\sum` u342
`\sup` B24
`\suppressfloats` [M1972](#)
`\supset` u417
`\supseteq` u419
`\surd` u328
`\sw@slant` w91, [w101](#)
`\swarrow` u400
`\swshape` w30
`\symbol` l162, [t293](#)
`\symletters` x8, x14, x49, x1052
`\symoperators` u635
- T**
- `\T` g23, l335,
 l337, l339, l341, l343, l345, l347,
 l349, l351, l374, N797, N801, N802
`\t` l282, l735, l843, x181, x182,
 x196, x198, x199, x211, x214,
 x216, x602, x772, x1039, x1041
`\t@st@ic` w90, [w94](#)
`\tabbing` [E59](#)
`\tabbingsep` E118, E120, [E154](#)
`\tabcolsep` E235, [E314](#)
- `\tabskip` b423, B195, B196,
 B322, B325, B328, B330, B441,
 B454, B457, B459, E155, E180
`\tabular` [E162](#)
`\tabular*` [E163](#)
`\tabularnewline` E182, [E195](#)
`\tan` B15
`\tanh` B17
`\target@meta@family@value`
 t123, t136, t141, t143
`\target@series@value` t122,
 t128, t131, t145, t147, t148, t149
`\tau` u277
`\tc@check@accent`
 x130, x184, x185, x186, x187,
 x188, x189, x190, x191, x192,
 x193, x194, x195, x196, x197,
 x198, x199, [x851](#), x927, x929, x931
`\tc@check@symbol` x130, x218,
 x219, x220, x221, x222, x223,
 x224, x225, x226, x227, x228,
 x229, x230, x231, x232, x233,
 x234, x235, x236, x237, x238,
 x239, x240, x241, x242, x243,
 x244, x245, x246, x247, x248,
 x249, x250, x251, x252, x253,
 x254, x255, x256, x257, x258,
 x259, x260, x261, x262, x263,
 x264, x265, x266, x268, x269,
 x270, x271, x272, x273, x274,
 x275, x276, x277, x278, x279,
 x280, x281, x282, x283, x284,
 x285, x286, x287, x288, x291,
 x292, x293, x294, x295, x296,
 x297, [x851](#), x921, x923, x925,
 x933, x935, x937, x939, x941,
 x943, x945, x947, x949, x951,
 x953, x955, x957, x959, x961,
 x963, x965, x967, x969, x971,
 x973, x975, x977, x979, x981,
 x983, x985, x987, x989, x991,
 x993, x995, x997, x999, x1001,
 x1003, x1005, x1007, x1009,
 x1011, x1013, x1015, x1017,
 x1019, x1021, x1023, x1025,
 x1027, x1029, x1031, x1033, x1035
`\tc@error` x131, [x831](#), x852
`\tc@errorwarn`
 ... x21, [x97](#), x99, x740, x741,
 x742, x743, x790, x791, x792, x825
`\tc@fake@euro` . [x118](#), x290, [x839](#), x920
`\tc@forcedfalse` x784
`\tc@forcedtrue` x789
`\tc@oldstylesubst` x16, x20

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\tc@subst` [x98](#), [x130](#), [x824](#), [x851](#)
`\tc@swap@accent` [x132](#), [x133](#)
`\tencirc` [v10](#), [F37](#), [F364](#)
`\tencircw` [v10](#), [F39](#)
`\tenln` [v9](#), [F37](#), [F38](#), [F363](#), [F365](#)
`\tenlnw` [v9](#), [F39](#), [F40](#)
`\TeX` [j1](#), [j12](#)
`\TeXOrMath` [m127](#), [m143](#)
`\text@command` [w8](#), [w46](#)
`\textacutedbl`
 [l898](#), [l1104](#), [x230](#), [x635](#), [x884](#)
`\textascendercompwordmark`
 [l848](#), [x178](#), [x599](#), [x867](#)
`\textasciicircum`
 [l948](#), [l1065](#), [x231](#), [x636](#), [x908](#)
`\textasciibreve`
 [l896](#), [l1103](#), [x232](#), [x637](#), [x881](#)
`\textasciicaron`
 [l897](#), [l1102](#), [x233](#), [x638](#), [x882](#)
`\textasciicircum` ... [l286](#), [l555](#), [l1039](#)
`\textasciidieresis`
 [l936](#), [l1052](#), [x234](#), [x639](#), [x898](#)
`\textasciigrave`
 [l887](#), [l1031](#), [x235](#), [x640](#), [x879](#)
`\textasciimacron`
 [l943](#), [l1060](#), [x236](#), [x641](#), [x903](#)
`\textasciitilde` [l287](#), [l556](#), [l1044](#)
`\textasteriskcentered`
 [l267](#), [l711](#), [l858](#), [l1157](#),
 [m130](#), [m136](#), [x144](#), [x507](#), [x562](#), [x874](#)
`\textbackslash` . [l268](#), [l557](#), [l712](#), [l1038](#)
`\textbaht` [l922](#),
 [l1107](#), [x239](#), [x645](#), [x1014](#), [x1015](#)
`\textbar` [l269](#), [l558](#), [l713](#), [l1042](#)
`\textbardbl`
 [l270](#), [l714](#), [l902](#), [l1112](#), [m135](#),
 [x151](#), [x297](#), [x508](#), [x570](#), [x711](#), [x887](#)
`\textbf` [w19](#)
`\textbigcircle` [l723](#),
 [l875](#), [l1174](#), [x240](#), [x646](#), [x966](#), [x967](#)
`\textblank`
 [l855](#), [l1171](#), [x286](#), [x699](#), [x936](#), [x937](#)
`\textborn`
 [l888](#), [x241](#), [x318](#), [x647](#), [x972](#), [x973](#)
`\textbraceleft`
 [l271](#), [l308](#), [l559](#), [l715](#), [l1041](#)
`\textbraceright`
 [l272](#), [l309](#), [l560](#), [l716](#), [l1043](#)
`\textbrokenbar`
 [l934](#), [l1050](#), [x152](#), [x571](#), [x896](#)
`\textbullet` [l273](#), [l717](#),
 [l904](#), [l1121](#), [x145](#), [x509](#), [x563](#), [x889](#)
`\textcapitalcompwordmark`
 [l847](#), [x177](#), [x598](#), [x866](#)
`\textcelsius` [l905](#),
 [l1140](#), [x153](#), [x291](#), [x572](#), [x705](#), [x890](#)
`\textcent` [l930](#), [l1046](#), [x154](#), [x573](#), [x893](#)
`\textcentoldstyle` [l907](#),
 [x242](#), [x323](#), [x326](#), [x648](#), [x988](#), [x989](#)
`\textcircled` . [l279](#), [l283](#), [l300](#), [l301](#),
 [l724](#), [l876](#), [x179](#), [x180](#), [x574](#),
 [x587](#), [x601](#), [x771](#), [x1036](#), [x1038](#)
`\textcircledP` [l941](#),
 [l1142](#), [x243](#), [x649](#), [x1030](#), [x1031](#)
`\textcolonmonetary`
 [l909](#), [l1133](#), [x270](#), [x679](#), [x990](#), [x991](#)
`\textcommaabove` [l353](#), [l355](#),
 [l369](#), [l370](#), [l455](#), [l456](#), [l694](#), [l695](#)
`\textcommabelow` [l324](#), [l326](#),
 [l332](#), [l333](#), [l697](#), [l698](#), [l699](#), [l700](#),
 [l701](#), [l702](#), [l703](#), [l704](#), [l705](#), [l706](#),
 [l1194](#), [l1391](#), [l1392](#), [l1393](#), [l1394](#)
`\textcompsubstdefault`
 [x34](#), [x39](#), [x53](#), [x829](#)
`\textcompwordmark` [l290](#), [l291](#), [l561](#), [l1109](#)
`\textcopyleft` [l939](#),
 [x244](#), [x299](#), [x650](#), [x1028](#), [x1029](#)
`\textcopyright` [l283](#),
 [l317](#), [l937](#), [l1053](#), [x155](#), [x574](#), [x899](#)
`\textcurrency` [l932](#), [l1048](#),
 [x281](#), [x692](#), [x766](#), [x770](#), [x924](#), [x925](#)
`\textdagger`
 [l275](#), [l312](#), [l719](#), [l900](#), [l1119](#),
 [m131](#), [m137](#), [x147](#), [x511](#), [x565](#), [x885](#)
`\textdaggerdbl`
 [l274](#), [l313](#), [l718](#), [l901](#), [l1120](#),
 [m132](#), [m138](#), [x146](#), [x510](#), [x564](#), [x886](#)
`\textdblhyphen`
 [l859](#), [x246](#), [x300](#), [x652](#), [x938](#), [x939](#)
`\textdblhyphenchar`
 [l895](#), [x245](#), [x301](#), [x651](#), [x984](#), [x985](#)
`\textdegree` [l944](#), [l1061](#), [x156](#), [x575](#), [x904](#)
`\textdied`
 [l890](#), [x247](#), [x319](#), [x653](#), [x976](#), [x977](#)
`\textdiscount` [l924](#),
 [l1132](#), [x248](#), [x654](#), [x1018](#), [x1019](#)
`\textdiv` . [l961](#), [l1086](#), [x157](#), [x576](#), [x918](#)
`\textdivorced`
 [l889](#), [l1177](#), [x249](#), [x655](#), [x974](#), [x975](#)
`\textdollar` .. [l255](#), [l307](#), [l435](#), [l562](#),
 [l792](#), [l856](#), [l1035](#), [x134](#), [x135](#),
 [x543](#), [x545](#), [x872](#), [x1044](#), [x1046](#)
`\textdollaroldstyle` [l906](#),
 [x250](#), [x324](#), [x325](#), [x656](#), [x986](#), [x987](#)
`\textdong` [l918](#),
 [l1137](#), [x271](#), [x680](#), [x1008](#), [x1009](#)
`\textdownarrow`
 [l886](#), [l1152](#), [x272](#), [x681](#), [x970](#), [x971](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\texteightoldstyle` 1869, x220, x315, x624, x956, x957
- `\textellipsis` 1296, 1321, 11122
- `\textemdash` 1256, 1407, 1563, 1779, 11111
- `\textendash` 1257, 1408, 1564, 1780, 11110
- `\textestimated` 1925, 11148, x278, x688, x769, x922, x923
- `\texteuro` 1959, 11138, x289, x290, x703, x767, x919, x920
- `\textexclamdown` 1258, 1409, 1411, 1565, 1781, 11045
- `\textfiveoldstyle` 1866, x221, x312, x625, x950, x951
- `\textfloatsep` M728, M741, M2104, M2154, M2303
- `\textflorin` 1908, 11100, x280, x691, x891
- `\textfont` q291, B205
- `\textfont@name` q285, q291
- `\textfouroldstyle` 1865, x222, x311, x626, x948, x949
- `\textfraction` M1917, M1920, M1944, M1947, M2096, M2297
- `\textfractionsolidus` 1860, 11129, x282, x694, x875
- `\textgravedbl` 1899, 11105, x237, x642, x883
- `\textgreater` ... 1281, 1566, 1734, 11037
- `\textguarani` 1912, x251, x322, x657, x996, x997
- `\textheight` k20, k21, k77, k78, M78, M225, M226, M274, M399, M447, M474, M645, M704, M763, M815, Q89, Q90, I257, I258, I261, I287, I301
- `\texthyphen` 1260, 1414, 1568, 1783
- `\texthyphenchar` . 1259, 1413, 1567, 1782
- `\textinterrobang` 1916, 11128, x288, x701, x1004, x1005
- `\textinterrobangdown` 1917, 11178, x287, x700, x1006, x1007
- `\textit` w21
- `\textlangle` 1871, 11169, x268, x676, x960, x961
- `\textlbrackdbl` 1883, x218, x320, x621, x877
- `\textleaf` 1891, x252, x304, x658, x978, x979
- `\textleftarrow` 1853, 11149, x273, x682, x932, x933
- `\textlegacyasteriskcentered` x519, x714
- `\textlegacybardbl` x519, x715
- `\textlegacybullet` x519, x716
- `\textlegacydagger` x519, x718
- `\textlegacydaggerdbl` x519, x717
- `\textlegacyparagraph` x519, x719
- `\textlegacyperiodcentered` x519, x720
- `\textlegacysection` x519, x721
- `\textless` 1280, 1569, 1733, 11036
- `\textlira` 1914, 11134, x274, x683, x1000, x1001
- `\textlnot` 1940, 11058, x158, x577, x901
- `\textlquill` 1928, 11130, x253, x659, x1024, x1025
- `\textmarried` 1892, 11176, x254, x660, x980, x981
- `\textmd` w19
- `\textmho` 1874, 11147, x255, x661, x964, x965
- `\textminus` 1872, 11153, x285, x697, x876
- `\textmu` .. 1949, 11066, x284, x696, x909
- `\textmusicalnote` 1893, 11175, x256, x662, x982, x983
- `\textnaira` 1911, 11135, x257, x663, x994, x995
- `\textnineoldstyle` 1870, x223, x316, x627, x958, x959
- `\textnormal` w15
- `\textnumero` 1923, 11141, x279, x689, x1016, x1017
- `\textogonekcentered` .. 1484, 1692, 1693
- `\textohm` 1882, 11146, x283, x695, x768, x921
- `\textonehalf` 1957, 11075, x159, x578, x915
- `\textoneoldstyle` 1862, x224, x308, x628, x942, x943
- `\textonequarter` 1956, 11074, x160, x579, x914
- `\textonesuperior` 1953, 11069, x161, x292, x580, x706, x912
- `\textopenbullet` 1926, 11173, x258, x664, x1020, x1021
- `\textordfeminine` 1305, 1938, 11054, x162, x581, x900
- `\textordmasculine` 1306, 1954, 11070, x163, x582, x913
- `\TextOrMath` m130, m131, m132, m133, m134, m135, m136, m137, m138, m150
- `\textparagraph` 1276, 1310, 1720, 1950, 11067, m134, x148, x512, x566, x910
- `\textperiodcentered` ... 1277, 1721, 1951, 11068, x149, x513, x567, x911
- `\textpertenthousand` 1489, 1920, 11124, x266, x267, x673, x1010, x1011, x1048

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\textperthousand` 1487, 1903, 11123,
 x138, x139, x559, x888, x1047
`\textpeso`
 1913, 11139, x259, x665, x998, x999
`\textpilcrow` 1921,
 x260, x317, x666, x1012, x1013
`\textpm` .. 1945, 11062, x164, x583, x905
`\textquestiondown`
 1261, 1410, 1412, 1570, 1784, 11077
`\textquotedbl` 1573, 11033
`\textquotedblleft`
 1262, 1415, 1571, 1785, 11116
`\textquotedblright`
 1263, 1416, 1572, 1786, 11117
`\textquoteleft`
 1264, 1417, 1574, 1787, 11113
`\textquoteright`
 1265, 1418, 1575, 1788, 11114
`\textquotesingle`
 1857, 11029, x165, x584, x873
`\textquotestraightbase`
 1849, x166, x302, x585, x868
`\textquotestraightdblbase`
 1850, x167, x303, x586, x869
`\texttriangle`
 1873, 11170, x269, x677, x962, x963
`\texttrbrackdbl`
 1884, x219, x321, x622, x878
`\textrecipe` 1915,
 11143, x261, x667, x1002, x1003
`\textreferencemark` 1952,
 11127, x262, x668, x1032, x1033
`\textregistered` 1300,
 1301, 1942, 11059, x168, x587, x902
`\textrightarrow`
 1854, 11151, x275, x684, x934, x935
`\textrm` w15
`\textrquill` 1929,
 11131, x263, x669, x1026, x1027
`\textsc` w21
`\textsection` 1278,
 1311, 1576, 1722, 1935, 11051,
 m133, x150, x514, x515, x568, x897
`\textservicemark` 1927,
 11144, x264, x670, x1022, x1023
`\textsevenoldstyle`
 1868, x225, x314, x629, x954, x955
`\textsf` w15
`\textsixoldstyle`
 1867, x226, x313, x630, x952, x953
`\textsl` w21
`\textssc` w25, w31, w39
`\textsterling` 1266, 1319, 1442, 1577,
 1799, 1931, 11047, x136, x137,
 x544, x552, x894, x1043, x1045
`\textstyle` j15, u461, B63
`\textsubscript` 1382, 1393, 1394
`\textsuperscript` 1303,
 1305, 1306, x581, x582, x593, 1378
`\textsurd` 1955,
 11168, x265, x671, x1034, x1035
`\textsw` w25, w30, w38
`\TextSymbolUnavailable` 13, 1752
`\textthreeoldstyle`
 1864, x227, x310, x631, x946, x947
`\textthreequarters`
 1958, 11076, x170, x590, x916
`\textthreequartersemdash` ... 1852,
 x169, x293, x306, x589, x707, x871
`\textthreesuperior` 1947,
 11064, x171, x294, x591, x708, x907
`\texttildelow`
 1894, 11106, x238, x643, x880
`\texttimes` 1960, 11080, x172, x592, x917
`\texttrademark`
 1303, 1919, 11145, x173, x593, x892
`\texttt` w15
`\texttwelveudash` 1851,
 x174, x295, x305, x594, x709, x870
`\texttwooldstyle`
 1863, x228, x309, x632, x944, x945
`\texttwosuperior` 1946,
 11063, x175, x296, x595, x710, x906
`\textulc` w25, w28, w29, w35, w37
`\textunderscore` 1288, 1315, 1578, 11040
`\textup` w21
`\textuparrow`
 1885, 11150, x276, x685, x968, x969
`\textvisiblespace` .. 1292, 1579, 11172
`\textwidth` k22,
 k79, D301, M79, M144, M201,
 M218, M630, M640, M689,
 M699, M2223, M2255, Q90, 1266
`\textwon`
 1910, 11136, x277, x686, x992, x993
`\textyen` . 1933, 11049, x176, x596, x895
`\textzerooldstyle`
 1861, x229, x307, x633, x940, x941
`\tf@size` o187, o207,
 o562, p187, p207, p562, q282, q284
`\TH` 1532, 11082, Q481
`\th` 1580, 11088, Q481
`\thanks` 441, H10, H26
`thebibliography` (environment) .. 474
`\theequation` . B306, B318, B377, B438
`\thefootnote` 1372, 1424, 1429, 1449

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \thempdfn [D303](#), [I400](#), [I405](#), [I440](#), [I445](#), [I448](#)
- \thempfootnote [D303](#), [I374](#)
- \thepage [k137](#),
[y6](#), [z14](#), [z34](#), [H164](#), [H171](#), [J15](#),
[J32](#), [K23](#), [L14](#), [M244](#), [M275](#), [M1827](#)
- \Theta [u291](#)
- \theta [u267](#)
- \thicklines [F37](#)
- \thickmuskip [u638](#), [B203](#)
- \thinlines [F37](#), [F434](#), [F451](#)
- \thinmuskip ... [i336](#), [u636](#), [B201](#), [B204](#)
- \thinspace [i336](#),
[i360](#), [i361](#), [i368](#), [i369](#), [B176](#), [B205](#)
- \thispagestyle [L6](#)
- \thr@@ . [b16](#), [b473](#), [q58](#), [q208](#), [q214](#),
[q227](#), [q234](#), [q241](#), [q246](#), [B329](#),
[B458](#), [C232](#), [C243](#), [F145](#), [F146](#),
[F148](#), [F149](#), [F181](#), [F203](#), [Q70](#), [Q78](#)
- \tilde [u511](#)
- \time [a179](#), [a183](#)
- \times [u388](#)
- \title ... [441](#), [H6](#), [H7](#), [H21](#), [H23](#), [H31](#)
- \to [u432](#), [u434](#)
- \today [a184](#), [a188](#), [a196](#), [a199](#), [H33](#)
- \toks [b31](#), [b63](#), [b95](#),
[s453](#), [s454](#), [s464](#), [s473](#), [P34](#), [Q532](#)
- \toks@ . [b41](#), [c81](#), [c85](#), [c88](#), [c93](#), [i318](#),
[i319](#), [i324](#), [o112](#), [o116](#), [o118](#),
[o121](#), [o185](#), [o190](#), [p112](#), [p116](#),
[p118](#), [p121](#), [p185](#), [p190](#), [s6](#), [s7](#),
[s259](#), [s263](#), [s269](#), [s272](#), [s277](#),
[s323](#), [s324](#), [s326](#), [s327](#), [s357](#),
[s359](#), [s363](#), [s380](#), [s383](#), [s442](#),
[s454](#), [s455](#), [s456](#), [s502](#), [s504](#),
[s510](#), [s518](#), [s522](#), [s534](#), [s537](#),
[s540](#), [s548](#), [s550](#), [s1035](#), [s1037](#),
[s1039](#), [s1042](#), [s1044](#), [s1047](#),
[s1050](#), [s1082](#), [s1083](#), [M2209](#),
[M2210](#), [M2211](#), [M2212](#), [N210](#),
[N211](#), [N213](#), [N214](#), [N506](#), [N507](#)
- \toksdef [b46](#), [b63](#), [b95](#), [P217](#)
- \tokszero [P217](#)
- \tolerance [b304](#), [o582](#), [o627](#),
[o642](#), [p582](#), [p627](#), [p642](#), [L58](#), [L66](#)
- \top [u312](#)
- \topfigrule [M727](#), [M2325](#)
- \topfraction [M2291](#), [I273](#)
- \topmargin [M71](#), [M624](#), [M683](#)
- \topmark [M2209](#), [M2218](#)
- \topsep [B443](#), [C2](#), [C59](#)
- \topskip ... [b343](#), [k56](#), [k111](#), [C1](#), [M128](#)
- \totalheight [D33](#), [D34](#), [D35](#)
- \tracefloats [M1897](#)
- \tracefloatsoff [M1897](#)
- \tracefloatvals [M1897](#)
- \tracingall [b446](#)
- \tracingassigns [b475](#), [b508](#)
- \tracingcommands
... [b457](#), [b473](#), [b482](#), [b494](#), [b511](#)
- \tracingfonts [q17](#), [q54](#), [q58](#),
[q86](#), [q116](#), [q125](#), [q148](#), [q178](#),
[q192](#), [q208](#), [q214](#), [q227](#), [q234](#),
[q241](#), [q246](#), [q255](#), [q268](#), [q276](#), [q279](#)
- \tracinggroups [b466](#), [b519](#)
- \tracingifs [b467](#), [b518](#)
- \tracinglostchars
[b320](#), [b452](#), [b464](#), [b483](#), [b502](#), [b522](#)
- \tracingmacros
... [b456](#), [b472](#), [b484](#), [b501](#), [b521](#)
- \tracingnesting [b469](#), [b516](#)
- \tracingnone [b488](#)
- \tracingoff [q116](#), [q276](#)
- \tracingon [q117](#), [q277](#)
- \tracingonline
[b441](#), [b493](#), [b510](#), [b531](#), [o591](#), [p591](#)
- \tracingoutput [b442](#), [b497](#), [b514](#), [b528](#)
- \tracingpages
... [b451](#), [b463](#), [b483](#), [b503](#), [b523](#)
- \tracingparagraphs
... [b453](#), [b465](#), [b484](#), [b500](#), [b520](#)
- \tracingrestores
... [b458](#), [b474](#), [b484](#), [b499](#), [b509](#)
- \tracingscantokens
... [b448](#), [b468](#), [b491](#), [b517](#)
- \tracingstats
[b450](#), [b462](#), [b482](#), [b504](#), [b524](#), [Q2](#)
- \triangle [u314](#)
- \triangleleft [u350](#), [u474](#)
- \triangleright [u351](#), [u474](#)
- \trivlist [A172](#), [A202](#), [A204](#), [A218](#),
[A240](#), [B428](#), [C89](#), [E66](#), [G35](#), [G37](#)
- \try@load@fontshape
... [o307](#), [o315](#), [o361](#), [o466](#), [p307](#),
[p315](#), [p361](#), [p466](#), [q474](#), [s208](#), [s225](#)
- \try@simple@size [q310](#), [q435](#)
- \try@simples [q393](#), [q399](#), [q403](#)
- \try@size@range [q101](#), [q310](#), [q386](#)
- \try@size@substitution .. [q103](#), [q390](#)
- \tryif@simple [q401](#), [q402](#)
- \tryis@simple [q402](#)
- \ttdef@ult [t72](#), [t83](#), [t95](#), [t167](#), [x31](#), [x109](#)
- \ttdefault [t12](#), [t72](#), [t113](#), [t162](#), [t203](#), [u48](#)
- \ttfamily [t10](#), [t11](#),
[t106](#), [t167](#), [t201](#), [t202](#), [w17](#), [A262](#)
- \ttsubstdefault .. [u19](#), [u30](#), [x32](#), [x110](#)
- \tw@ [b16](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\two@digits` a86, a185,
 a186, [d2](#), q466, [N559](#), N655, N745
`\twocolumn` [M199](#)
`\type@restoreinfo` q156, [q161](#)
`\typein` 37, [37](#), [d16](#)
`\typeout` 37, a73, a116, a172,
 a197, a199, a211, a226, a233,
 a244, a257, a270, a283, a321,
 c21, c38, c43, c48, [d3](#), d21, d28,
 g74, k129, k358, k359, k361,
 k396, k406, k409, o301, p301,
 t238, t341, t351, t361, u9, u117,
 J8, J25, M1898, N158, N173,
 N814, N1013, N1016, Q207,
 Q485, Q492, Q504, Q505, Q513
- U**
- `\u` l237, l386, l469, l583,
 l590, l610, l617, l748, l1184,
 l1259, l1260, l1275, l1276, l1285,
 l1286, l1299, l1300, l1301, l1325,
 l1326, l1351, l1352, x185, x202
`\uccode` Q173, Q181,
 Q188, Q190, Q193, Q195, Q436,
 Q444, Q451, Q453, Q456, Q458
`\uchyph` b321
`\ulcshape` t224, w29
`\Umathchar` .. u15, u48, u63, P28, Q247
`\Umathcode`
 b127, x183, x298, A268, Q93, Q430
`\unboldmath` t290
`\UndeclareTextCommand` [l191](#), x135,
 x137, x139, x267, x515, x1043,
 x1044, x1045, x1046, x1047, x1048
`\undefined` . a9, a11, a17, a57, m92,
 m93, m94, m95, m96, m97,
 m98, m99, Q101, Q113, Q114, Q134
`\undefinedpagestyle` L4, L8
`\underbar` [b411](#), d438, d459
`\underbrace` u532
`\underline` ... 380, [b411](#), D358, [D359](#)
`\unexpanded` .. A102, A119, N842, N844
`\unhcopy` [b413](#), E321, F418
`\unicodedataline`
 .. P143, P146, P160, P161, P162
`\UnicodeEncodingName` .. 1964, 1970,
 1998, l1009, l1013, l1029, l1031,
 l1033, l1035, l1036, l1037, l1038,
 l1039, l1040, l1041, l1042, l1043,
 l1044, l1045, l1046, l1047, l1048,
 l1049, l1050, l1051, l1052, l1053,
 l1054, l1055, l1057, l1058, l1059,
 l1060, l1061, l1062, l1063, l1064,
 l1065, l1066, l1067, l1068, l1069,
 l1070, l1071, l1073, l1074, l1075,
 l1076, l1077, l1078, l1079, l1080,
 l1081, l1082, l1083, l1084, l1085,
 l1086, l1087, l1088, l1089, l1090,
 l1091, l1092, l1093, l1094, l1095,
 l1096, l1097, l1098, l1099, l1100,
 l1101, l1102, l1103, l1104, l1105,
 l1106, l1107, l1108, l1109, l1110,
 l1111, l1112, l1113, l1114, l1115,
 l1116, l1117, l1118, l1119, l1120,
 l1121, l1122, l1123, l1124, l1125,
 l1126, l1127, l1128, l1129, l1130,
 l1131, l1132, l1133, l1134, l1135,
 l1136, l1137, l1138, l1139, l1140,
 l1141, l1142, l1143, l1144, l1145,
 l1146, l1147, l1148, l1149, l1150,
 l1151, l1152, l1153, l1155, l1156,
 l1157, l1168, l1169, l1170, l1171,
 l1172, l1173, l1174, l1175, l1176,
 l1177, l1178, l1179, l1180, l1181,
 l1182, l1183, l1184, l1185, l1186,
 l1187, l1188, l1189, l1190, l1191,
 l1192, l1193, l1194, x307, x308,
 x309, x310, x311, x312, x313,
 x314, x315, x316, x317, x318,
 x319, x320, x321, x322, x323, x324
`\UnicodeFontFile` 1996
`\UnicodeFontName` 1997
`\UnicodeFontTeXLigatures` .. 1976, 1993
`\unicoderead` P143,
 P157, P158, P159, P160, P165
`\uninstall` [P773](#)
`\unitlength` ... D49, D59, [F5](#), F12,
 F13, F14, F15, F22, F23, F26,
 F27, F34, F58, F116, F169,
 F171, F184, F189, F191, F206,
 F208, F211, F251, F252, F296,
 F297, F331, F342, F372, F373,
 F375, F376, F379, F380, F382,
 F383, F392, F394, F396, F398, Q82
`\unkern` o610, p610
`\unless` P151, P159, P161
`\unlhd` t330
`\unpenalty` o613, o617,
 p613, p617, w116, A234, A256
`\unquote@name` k230, [k236](#), k275
`\unrestored@protected@xdef`
 . [d227](#), L25, L49, I405, I429, I445
`\unrhd` t332
`\unsetattribute` 589, [P82](#), P227
`\unvcopy` B180
`\Uparrow` u576
`\uparrow` u570
`\upbracefill` u535, u552

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
 t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
 y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
 D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
 I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
 N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

<code>\update@series@target@value</code>	<code>\UTFviii@undefined@err</code>
t124, t134, t210	Q302, Q307, Q313
<code>\update@ucrlc@with@cyrillic</code>	<code>\UTFviii@undefined@err@@</code> Q302, Q313
l1441, l1469, l1493, l1500	
<code>\updefault</code>	V
. t21, u92, u99, u100, u105, u107	<code>\v</code> l238, l387, l468, l586, l587, l588,
<code>\Updownarrow</code> u580	l592, l594, l597, l599, l601, l607,
<code>\updownarrow</code> u574	l613, l614, l615, l619, l621, l624,
<code>\uplus</code> u370	l626, l628, l634, l749, l1189,
<code>\upper@bound</code> q337, q338, q339, q352	l1269, l1270, l1271, l1272, l1281,
<code>\uppercase</code> Q468	l1282, l1315, l1316, l1321, l1322,
<code>\upshape</code>	l1333, l1334, l1341, l1342, l1345,
l439, l726, l796, l878, t19, t20,	l1346, l1368, l1369, l1370, l1371,
t224, t235, t268, t309, w24, x549	l1372, l1373, l1374, l1375, l1376,
<code>\Upsilon</code> u296	l1377, l1378, l1381, l1382, l1383,
<code>\upsilon</code> u278	l1384, l1387, l1388, x186, x203
<code>\use@mathgroup</code> o530, o548, o550,	<code>\v@false</code> B82
p530, p548, p550, q253, s63,	<code>\v@true</code> B81, B83
s92, s424, s526, s529, s1060, s1084	<code>\vadjust</code> i37,
<code>\usebox</code> D111	i59, i71, i83, i287, i303, l201, l223
<code>\usecounter</code> C225, C238	<code>\valign</code> x122, x843
<code>\usefont</code> l1499, o44, o246, o587, p44,	<code>\value</code> l45, m14, K9
p246, p587, t312, x7, x48, A272	<code>\varbigtriangledown</code> u354, u357
<code>\UseLegacyTextSymbols</code> x506, x713	<code>\varbigtriangleup</code> u355, u356
<code>\usepackage</code> N302, N332, N829	<code>\varepsilon</code> j15, u283
<code>\UseRawInputEncoding</code> Q277, Q333, Q380	<code>\varphi</code> u288
<code>\UseTextAccent</code> l149, l150, l188,	<code>\varpi</code> u285
x131, x132, x133, x180, x182,	<code>\varrho</code> u286
x852, x1037, x1038, x1040, x1041	<code>\varsigma</code> u287
<code>\UseTextSymbol</code>	<code>\vartheta</code> u284
l150, l186, x130, x290, x851, x920	<code>\vbadness</code> b306, M2206
<code>\UTF@four@octets@noexpand</code> Q467, Q476	<code>\vdash</code> u396
<code>\UTF@three@octets@noexpand</code>	<code>\vdots</code> u502
Q466, Q475	<code>\vec</code> u516
<code>\UTF@two@octets@noexpand</code> Q465, Q474	<code>\vector</code> g219, F113, F435, F452
<code>\UTFviii@four@octets</code> Q306, Q311, Q317	<code>\vee</code> u359, u361
<code>\UTFviii@four@octets@@</code> Q306, Q317	<code>\verb</code> A288, A314, A327, A329
<code>\UTFviii@four@octets@combine</code> Q341	<code>\verb@balance@group</code> A287,
<code>\UTFviii@four@octets@noexpand</code> Q347	A289, A313, A315, A321, A322
<code>\UTFviii@four@octets@string</code> Q344	<code>\verb@egroup</code> A287,
<code>\UTFviii@invalid</code> Q245, Q338	A290, A313, A316, A322, A326
<code>\UTFviii@invalid@err</code> Q303, Q308, Q314	<code>\verb@eol@error</code> A323, A335, A345
<code>\UTFviii@invalid@err@@</code> Q303, Q314	<code>\verbatim</code> A260
<code>\UTFviii@three@octets</code>	<code>verbatim*</code> (environment) A294
Q305, Q310, Q316	<code>\verbatim@font</code>
<code>\UTFviii@three@octets@@</code> Q305, Q316 A233, A254, A262, A336, A346
<code>\UTFviii@three@octets@combine</code> Q340	<code>\verbatim@nolig@list</code> A351, A357
<code>\UTFviii@three@octets@noexpand</code> Q346	<code>\verb@visible@space</code> A266,
<code>\UTFviii@three@octets@string</code> Q343	A268, A275, A279, A301, A306
<code>\UTFviii@two@octets</code> Q304, Q309, Q315	<code>\version@elt</code> s18, s31, s32, s256, s257,
<code>\UTFviii@two@octets@@</code> Q304, Q315	s306, s326, s417, s455, s547, s1040
<code>\UTFviii@two@octets@combine</code> Q339	<code>\version@list</code> s16,
<code>\UTFviii@two@octets@noexpand</code> Q345	s21, s32, s249, s257, s311, s332,
<code>\UTFviii@two@octets@string</code> Q342	s351, s422, s467, s497, s552, s1053

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx, t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx, y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx, D=ltboxes.dtx, E=lttab.dtx, F=lt pictur.dtx, G=ltthm.dtx, H=ltsect.dtx, I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx, N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\Vert` [u563](#), [u565](#)
`\vert` [u568](#)
`\vfil` [b400](#), [x123](#), [x126](#),
[x844](#), [x847](#), [F275](#), [F318](#), [M175](#),
[M194](#), [M412](#), [M459](#), [M627](#), [M686](#)
`\vfilneg` [b400](#)
`\vfuzz` [b329](#), [L61](#), [L68](#)
`\vgl@` [b390](#), [b391](#)
`\vglue` [b390](#)
`\vline` [E342](#)
`\voidb@x` [b298](#), [b421](#), [n18](#)
`\vphantom` [l494](#), [l510](#), [B75](#)
`\vrule` [b394](#), [i352](#), [l293](#), [l295](#),
[l499](#), [l515](#), [q144](#), [u550](#), [u551](#),
[u553](#), [u554](#), [D120](#), [D122](#), [D173](#),
[D180](#), [D357](#), [D401](#), [E174](#), [E207](#),
[E323](#), [E342](#), [F107](#), [F157](#), [F160](#),
[F176](#), [F183](#), [F198](#), [F205](#), [F274](#),
[F318](#), [F402](#), [M1851](#), [M2226](#), [M2259](#)
`\vspace` [i280](#), [i310](#), [i311](#), [i312](#)
`\vsplit` [M382](#), [M429](#), [M2208](#)
- W**
- `\warn@rel@i` [r5](#), [r25](#),
[r29](#), [r81](#), [r85](#), [r90](#), [r95](#), [r119](#), [r140](#)
`\wedge` [u358](#), [u360](#)
`\whatsit` [P185](#)
`\widehat` [u519](#)
`\widetilde` [u518](#)
`\widowpenalties` [b106](#)
`\widowpenalty` [b313](#), [o600](#), [p600](#)
`\width` [D30](#)
`\wlog` [a100](#), [b40](#),
[b145](#), [b230](#), [b243](#), [b273](#), [b288](#),
[N175](#), [P6](#), [P7](#), [P8](#), [P52](#), [Q46](#), [Q539](#)
`\wp` [u304](#)
`\wr` [u374](#)
`\wrong@fontshape` [o311](#), [o448](#), [p311](#), [p448](#)
- X**
- `\x` [o268](#), [o269](#), [p268](#), [p269](#)
`\x@protect` ... [d205](#), [d216](#), [d264](#), [d291](#)
`\xe@alloc@` [Q42](#), [Q52](#)
`\xe@alloc@intercharclass` [Q21](#)
`\xe@ch@ck` [Q43](#), [Q47](#)
- XeTeXcharclass**
- `\XeTeXcharclass` . [o575](#), [p575](#), [Q25](#),
[Q33](#), [Q40](#), [Q53](#), [Q59](#), [Q68](#), [Q75](#)
`\XeTeXcharclassCL` [Q106](#)
`\XeTeXcharclassCM` [Q110](#)
`\XeTeXcharclassEX` [Q107](#)
`\XeTeXcharclassID` [Q104](#)
`\XeTeXcharclassIS` [Q108](#)
`\XeTeXcharclassNS` [Q109](#)
`\XeTeXcharclassOP` [Q105](#)
`\XeTeXcharglyph` [l994](#)
`\XeTeXdashbreakstate` [Q204](#)
`\XeTeXglyph` [l994](#)
`\XeTeXintercharclasses` . [Q100](#), [Q133](#)
`\XeTeXinterchartoks` . [Q101](#), [Q115](#),
[Q116](#), [Q117](#), [Q118](#), [Q119](#), [Q120](#),
[Q121](#), [Q122](#), [Q123](#), [Q124](#), [Q125](#),
[Q126](#), [Q127](#), [Q128](#), [Q129](#), [Q134](#),
[Q139](#), [Q140](#), [Q141](#), [Q142](#), [Q143](#),
[Q144](#), [Q145](#), [Q146](#), [Q147](#), [Q148](#),
[Q149](#), [Q150](#), [Q151](#), [Q152](#), [Q153](#)
`\XeTeXmathcode` [Q94](#), [Q431](#)
`\XeTeXrevision` [Q27](#)
`\XeTeXuseglyphmetrics` .. [Q201](#), [Q203](#)
`\XeTeXversion` [Q27](#)
`\Xi` [u293](#)
`\xi` [u273](#)
`\xtxHanGlue` [Q113](#),
[Q137](#), [Q145](#), [Q146](#), [Q147](#), [Q148](#),
[Q149](#), [Q150](#), [Q151](#), [Q152](#), [Q153](#)
`\xtxHanSpace` .. [Q114](#), [Q138](#), [Q139](#),
[Q140](#), [Q141](#), [Q142](#), [Q143](#), [Q144](#)
- Y**
- `\year` [a185](#), [c11](#), [c14](#), [N655](#), [N745](#)
`\yxdim` [F354](#)
- Z**
- `\Z` [Q185](#), [Q427](#), [Q448](#)
`\z` [Q176](#), [Q428](#), [Q439](#)
`\z@` [b298](#)
`\z@skip` [b298](#)
`\zap@space` [k152](#), [k169](#),
[t221](#), [K29](#), [N108](#), [N199](#), [N281](#),
[N365](#), [N386](#), [N403](#), [N414](#), [N431](#)
`\zeta` [u265](#)

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=ltfssbas.dtx, q=ltfssstrc.dtx, r=ltfssscmp.dtx, s=ltfssdcl.dtx,
t=ltfssini.dtx, u=fontdef.dtx, v=preload.dtx, w=ltfntcmd.dtx, x=lttextcomp.dtx,
y=ltpageno.dtx, z=ltxref.dtx, A=ltmiscen.dtx, B=ltmath.dtx, C=ltlists.dtx,
D=ltboxes.dtx, E=lttab.dtx, F=ltpictur.dtx, G=ltthm.dtx, H=ltsect.dtx,
I=ltfloat.dtx, J=ltidxglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx