

The L^AT_EX 2_ε Sources

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

2020-02-02 Patch level 5

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	ltplain.dtx	15
9	Plain T_EX	15
c	ltxvers.dtx	33
10	Version Identification	33
d	ltxdefns.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	ltxalloc.dtx	54
13	Counters	54
f	ltxcntrl.dtx	56
14	Program control structure	56
g	ltxerror.dtx	60
15	Error handling and tracing	60
15.1	General commands	60
15.2	Specific errors	65
15.3	Tracing	68
h	ltxpar.dtx	69
16	Paragraphs	69
16.1	Implementation	69

i	itspace.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	133
21	Package files	142
21.1	The fontenc package	143
m	ltcounts.dtx	146
22	Counters and Lengths	146
22.1	Environment Counter Macros	146

n	ltnlength.dtx	153
23	Lengths	153
o	ltfssbas.dtx	154
24	Preliminary macros	154
25	Macros for setting up the tables	155
26	Selecting a new font	161
26.1	Macros for the user	161
26.2	Macros for loading fonts	165
27	Assigning math fonts to <i>versions</i>	172
p	ltfssaxes.dtx	178
28	Changing the font series	178
28.1	The series lookup table	178
28.2	Mapping rules for series changes	179
28.3	Changing to a new series	187
29	Changing the shape	189
29.1	Mapping rules for shape combinations	190
29.2	Changing to a new shape	192
30	Make sure we win . . .	193
q	ltfssstrc.dtx	195
31	Introduction	195
32	A driver for this document	195
33	The Implementation	196
34	Handling Options	196
35	Macros common to fam.tex and tracefnt.sty	198
35.1	General font loading	198
35.2	Math fonts setup	202
35.2.1	Outline of algorithm for math font sizes	202
35.2.2	Code for math font size setting	204
35.2.3	Other code for math	205
36	Scaled font extraction	207
36.1	Sizefunctions	214

r	lffscmp.dtx	218
s	lffsdcl.dtx	222
37	Interface Commands	222
t	lffssini.dtx	247
38	NFSS Initialisation	247
38.1	Providing math <i>versions</i>	247
39	Custom series settings for main document families	248
40	Supporting nested emphasis	257
40.1	Miscellaneous	260
u	fontdef.dtx	264
41	Introduction	264
42	Customization	264
43	The docstrip modules	265
44	A driver for this document	265
45	The fonttext.ltx file	266
45.1	Encodings	266
45.2	Defaults	268
46	The fontmath.ltx file	270
46.1	The font encodings used	270
46.1.1	Symbolfont and Alphabet declarations	270
46.2	Math font sizes	271
46.3	The math symbol assignments	271
46.3.1	The letters	272
46.3.2	The digits	273
46.3.3	Punctuation, brace, etc. keys	273
46.3.4	Delimitercodes for characters	273
46.4	Symbols accessed via control sequences	274
46.4.1	Greek letters	274
46.4.2	Ordinary symbols	275
46.4.3	Large Operators	275
46.4.4	Binary symbols	276
46.4.5	Relations	277
46.4.6	Arrows	278
46.4.7	Punctuation symbols	279
46.4.8	Math accents	279
46.4.9	Radicals	280

46.4.10	Over and under something, etc	280
46.4.11	Delimiters	281
46.5	Math versions of text commands	282
46.6	Other special functions and parameters	282
46.6.1	Biggggg	282
46.6.2	The log-like functions	283
46.6.3	Parameters	283
47	Default cfg files	283
v	preload.dtx	284
48	Overview	284
49	Customization	284
50	Module switches for the DOCSTRIP program	285
51	A driver for this document	285
52	The code	285
w	lftncmd.dtx	288
53	Introduction	288
54	The implementation	290
55	Initialization	296
x	ltxtextcomp.dtx	297
56	Sub-encodings	301
56.1	Sub-encoding 1 (drop symbols not working in Latin Modern)	302
56.2	Sub-encoding 2 (majority of new OTF fonts via autoinst)	303
56.3	Sub-encoding 3	305
56.4	Sub-encoding 4	305
56.5	Sub-encoding 5 (most older PS fonts)	305
56.6	Sub-encoding 6	305
56.7	Sub-encoding 7	305
56.8	Sub-encoding 8	306
56.9	Sub-encoding 9 (most missing)	306
57	Unicode engine specials	306
58	Font family sub-encodings setup	307
59	Legacy symbol support for lists and footnote symbols	311

60 The textcomp package	315
60.1 The old textcomp package code	316
60.1.1 Supporting oldstyle digits	325
60.1.2 Subset encoding defaults	325
y ltpageno.dtx	328
61 Page Numbering	328
z ltxref.dtx	329
62 Cross Referencing	329
62.1 Cross Referencing	329
A ltmiscen.dtx	333
63 Miscellaneous Environments	333
63.1 Environments	333
63.2 Center, Flushright, Flushleft	341
63.3 Verbatim	343
B ltmath.dtx	348
64 Math setup	348
64.1 Math commands based on plain T _E X	348
64.1.1 The log-like functions	348
64.1.2 Biggggg	349
64.1.3 The UNSORTED Rest	349
64.2 Math Environments	354
64.3 External options to the standard document classes	357
64.3.1 Left equation numbering	357
64.3.2 Flush left equations	358
C ltlists.dtx	361
65 List, and related environments	361
65.1 List and Trivlist	362
65.2 Vertical Spacing (skips)	363
65.3 Penalties	363
65.4 Horizontal Spacing (dimens)	363
65.5 Default Values	364
65.6 Itemize and Enumerate	374
D ltboxes.dtx	376

66	LaTeX Box commands	376
66.1	Some low-level constructs	388
E	ltablex.dtx	389
67	Tabbing, Tabular and Array Environments	389
67.1	tabbing	389
67.2	array and tabular environments	397
F	ltpictur.dtx	411
68	Picture Mode	411
68.1	Curves	431
G	lthm.dtx	434
69	Theorem Environments	434
H	ltssect.dtx	438
70	Sectioning Commands	438
70.1	The Title	438
70.2	Sectioning	439
70.2.1	Initializations	445
70.3	Table of Contents etc.	446
70.3.1	Convention	446
70.3.2	Commands	446
I	lfloat.dtx	450
71	Floats	450
71.1	Floating Environments	450
71.2	Footnotes	463
J	ltxglo.dtx	469
72	Index and Glossary Generation	469
K	ltxbibl.dtx	471
73	Bibliography Generation	471
73.1	Default definitions	474
L	ltxpage.dtx	475

74 Page styles and related commands	475
74.1 Page Style Commands	475
74.2 How a page style makes running heads and feet	475
74.3 marking conventions	475
M ltoutput.dtx	478
75 Output Routine	478
75.1 Floats	478
75.1.1 Kludgings	533
75.1.2 Float control	534
75.1.3 Float placement parameters	546
N Itclass.dtx	550
76 Introduction	550
77 User interface	550
77.1 Option processing	551
78 Class and Package interface	552
78.1 Class name and version	552
78.2 Package name and version	552
78.3 Requiring other packages	552
78.4 Declaring new options	553
78.5 Safe Input Macros	554
79 Implementation	554
79.1 Hooks	567
79.2 Providing shipment	568
80 Package/class rollback mechanism	574
81 After Preamble	582
O lthyphen.dtx	583
P Itlualatex.dtx	585
82 Overview	585
83 Core T_EX functionality	585
84 Plain T_EX interface	586

85 Lua functionality	586
85.1 Allocators in Lua	586
85.2 Lua access to \TeX register numbers	587
85.3 Module utilities	588
85.4 Callback management	588
86 Implementation	589
86.1 Minimum Lua \TeX version	589
86.2 Older \LaTeX /Plain \TeX setup	589
86.2.1 Fixes to <code>etex.src/etex.sty</code>	590
86.2.2 luatex specific settings	590
86.3 Attributes	591
86.4 Category code tables	591
86.5 Named Lua functions	593
86.6 Custom whatsits	594
86.7 Lua bytecode registers	594
86.8 Lua chunk registers	594
86.9 Lua loader	594
86.10 Lua module preliminaries	596
86.11 Lua module utilities	596
86.11.1 Module tracking	596
86.11.2 Module messages	597
86.12 Accessing register numbers from Lua	598
86.13 Attribute allocation	599
86.14 Custom whatsit allocation	599
86.15 Bytecode register allocation	600
86.16 Lua chunk name allocation	600
86.17 Lua function allocation	601
86.18 Lua callback management	601
86.18.1 Housekeeping	601
86.18.2 Handlers	603
86.18.3 Public functions for callback management	606
Q ltfinal.dtx	612
87 Final settings	612
87.1 Debugging	612
87.2 Typesetting parameters	612
87.3 Lccodes for hyphenation	614
87.4 Hyphenation	617
87.5 Font loading	618
87.6 Input encoding	618
87.7 Lccodes and uccodes	622
87.8 Applying Patch files	624
87.9 Loading further extensions	625
87.10 Freeing Memory	625
87.11 Initialise file list	626
87.12 Some last minute initializations ...	626
87.13 Do some temporary work for pre-release	626

87.14 Dumping the format	626
Change History	627
Index	690

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 </initex, latexrelease>
53 (*dircheck)
54 (*initex)
55 \fi
56 \fi

```

A test can now be made for eTeX.

```

57 (initex) \ifx eTeXversion undefined
58 (initex) \errmessage
59 (initex) {LaTeX requires e-TeX}
60 (initex) \expandafter \endinput
61 (initex) \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 \@inputcheck 0
71 \chardef \sist@n=16
72 \newlinechar '\ ^^J
73 \def \typeout {\immediate \write 17}
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@tempswa \iffalse}
79 \def \@tempwattrue {\let \if@tempswa \iftrue}
80 \let \if@tempswa \iffalse
81 \def \loop #1 \repeat {\def \iterate {#1 \relax \expandafter \iterate} \fi}%
82 \iterate \let \iterate \relax}
83 \let \repeat \fi
84 </initex>

```

2.2 Some bits of 2e

```

85 (*2kernel)
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` $\langle filename \rangle \langle space \rangle$ 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 $\langle space \rangle$. 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 $\langle dir \rangle$ is an entry in the input path, \TeX will try to load the expansion of

$\langle dir \rangle \langle filename \rangle \langle space \rangle$

So either $\langle dir \rangle$ 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 $\langle filename \rangle$. This means that for UNIX-like syntax, each $\langle dir \rangle$ 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{\langle filename \rangle}`, 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 $\langle filename \rangle$, `\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 \enddocstrip)
```

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 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       \if\reserved@a\@empty\else\gdef\@TeXversion{3}\fi}
334   \fi
335 \fi

336 (/dircheck)

```

8 ltxcheck.tex

After the format has been made, and `article.cls` moved with the other files to the ‘standard input directory’ as specified in `install.txt`, the format may be checked

by running the file `ltxcheck.tex`.

File b

lplain.dtx

9 Plain T_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 (*2kernel)
2 \catcode'\{=1 % left brace is begin-group character
3 \catcode'\}=2 % right brace is end-group character
4 \catcode'\$=3 % dollar sign is math shift
5 \catcode'\&=4 % ampersand is alignment tab
6 \catcode'\#=6 % hash mark is macro parameter character
7 \catcode'\^=7 % circumflex and uparrow are for superscripts
8 \catcode'\_ =8 % underline and downarrow are for subscripts
9 \catcode'\^I=10 % ascii tab is a blank space
10 \chardef\active=13 \catcode'\^=\active % tilde is active
11 \catcode'\^L=\active \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 File b: ltplain.dtx Date: 2017/04/10 Version v2.3c
\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 </2kernel>
48 <*2kernel | latexrelease>
49 <latexrelease>\IncludeInRelease{2015/01/01}%
50 <latexrelease>          {\newcount}{Extended Allocation}%

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

For compatibility use \chardef in the classical range.

56 \def\newbox   {\e@alloc\box
57             {\ifnum\allocationnumber<\cclvi
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\cclvi}
83 \else
84 \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne{16384}}
85 \fi
86 </2kernel | 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

```

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

```

102 /*2kernel | latexrelease)
103 <latexrelease>\IncludeInRelease{2015/01/01}%
104 <latexrelease>                {\e@alloc@chardef}{Extended Allocation}%
105 \ifx\directlua\@undefined
106 \ifx\widowpenalties\@undefined
classic tex has 28 registers.
107 \mathchardef\e@alloc@top=255
108 \let\e@alloc@chardef\chardef
109 \else
etex and xetex have 215 registers.
110 \mathchardef\e@alloc@top=32767
111 \let\e@alloc@chardef\mathchardef
112 \fi
113 \else
luatex has 216 registers.
114 \chardef\e@alloc@top=65535
115 \let\e@alloc@chardef\chardef
116 \fi
117 </2kernel | latexrelease)
118 <latexrelease>\EndIncludeInRelease
119 <latexrelease>\IncludeInRelease{0000/00/00}%
120 <latexrelease>                {\e@alloc@chardef}{Extended Allocation}%
121 <latexrelease>\let\e@alloc@top\@undefined
122 <latexrelease>\let\e@alloc@chardef\@undefined
123 <latexrelease>\EndIncludeInRelease

```

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

```

124 /*2kernel | latexrelease)
125 <latexrelease>\IncludeInRelease{2015/01/01}%
126 <latexrelease>                {\e@mathgroup@top}{Extended Allocation}%
127 \ifx\Umathcode\@undefined
classic and e tex have 16 fam (0–15).
128 \chardef\e@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 \@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

```

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

```

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> \e@ch@ck}{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\@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 T_EX's parameters

```
302 \message{parameters,}
```

All of T_EX'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 L^AT_EX 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 \defaultshyphenchar='\-
323 \defaultskewchar=-1

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

324 \delimitfactor=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 \abovedisplaysshortskip=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 L^AT_EX 2.09 comments.

`\normalbaselineskip` We also define special registers that function like parameters:

```

\normalbaselineskip 346 \newskip\normalbaselineskip \normalbaselineskip=12pt
\normallineskip      347 \newskip\normallineskip \normallineskip=1pt
\normallineskiplimit 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 ltspc.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\allowbreak{\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`

```
\@lign 424 \def\oalign#1{\leavevmode \vtop{\baselineskip \z@skip \lineskip .25ex%
```

```
\oalign 425   \ialign{##\cr cr#1\cr cr}}}
```

```
426 \def\@lign{\lineskiplimit \z@ \oalign}
```

```
427 \def\oalign{\lineskiplimit-\maxdimen \oalign}
```

`\sh@ft`

The definition of this macro in `plain.tex` was improved in about 1997; but as a result its usage was changed and its new definition is not appropriate for L^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` L^AT_EX change: `\kern\z@` added to end of `\hrulefill` and `\dotfill` to make them work in ‘tabular’ and ‘array’ environments. (Change made 24 July 1987). L^AT_EX 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 L^AT_EX.

```
\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 /2ekernel)
```

`\tracingall`

```
\loggingall 446 (latexrelease)\IncludeInRelease{2015/01/01}{\loggingall}{etex tracing}%
```

```
447 (*2ekernel | 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

ltxvers.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).

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 {2020-02-02}
8 </2ekernel|latexrelease)
9 (*2ekernel)
10 \def\patch@level{5}
```

`\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 us internally when we locally install a format out of some development branch.

```
11 \edef\development@branch@name{}
```

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#2}

80 \def\@IncludeInRelease#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

ltdfn.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` saves 5 tokens at the cost in time of one macro expansion.

```
\@depth 11 \def\@height{height} \def\@depth{depth} \def\@width{width}
\@width 12 \def\@minus{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 <code>i</code> arguments (<code>i = 0</code> 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\newcommand}
```

`\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; `\@yargdef` 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   \@yargdef \expandafter{\number #3}#1%
90 }

91 \long \def \@yargdef#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\provide@command}
```

`\provide@command`

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

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

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

`\check@command`

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

`\@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   \renew@command\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{}
    \@gobblethree 184 \long\def \@gobbletwo #1#2{}
    \@gobblefour 185 \long\def \@gobblethree #1#2#3{}
    186 \long\def \@gobblefour #1#2#3#4{}

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

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

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

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

    \@backslashchar A category code 12 backslash.
    194 \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 `\prootected@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`

```
195 \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 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`.

```
196 \def\DeclareRobustCommand{\@star@or@long\declare@robustcommand}
```

```
197 \def\declare@robustcommand#1{%
198   \ifx#1\undefined\else\ifx#1\relax\else
199     \@latex@info{Redefining \string#1}%
200   \fi\fi
201   \edef\reserved@a{\string#1}%
202   \def\reserved@b{#1}%
203   \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
204   \edef#1{%
```

```

205     \ifx\reserved@a\reserved@b
206         \noexpand\x@protect
207         \noexpand#1%
208     \fi
209     \noexpand\protect
210     \expandafter\noexpand\csname
211         \expandafter\@gobble\string#1 \endcsname
212 }%
213 \let\@ifdefinable\@rc@ifdefinable
214 \expandafter\new@command\csname
215     \expandafter\@gobble\string#1 \endcsname
216 }

```

\@x@protect

```

\@x@protect
\@x@protect 217 \def\x@protect#1{%
218     \ifx\protect\@typeset@protect\else
219         \@x@protect#1%
220     \fi
221 }
222 \def\@x@protect#1\fi#2#3{%
223     \fi\protect#1%
224 }

```

\@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.

```
225 \let\@typeset@protect\relax
```

\set@display@protect These macros set \protect appropriately for typesetting or displaying.

```

\set@display@protect
\set@typeset@protect 226 \def\set@display@protect{\let\protect\string}
227 \def\set@typeset@protect{\let\protect\@typeset@protect}

```

\protected@edef The commands \protected@edef and \protected@xdef perform ‘safe’ \edefs and \xdefs, saving and restoring \protect appropriately. For cases where restoring \protect doesn’t matter, there’s an ‘unsafe’ \unrestored@protected@xdef, useful if you know what you’re doing!

\unrestored@protected@xdef
\restore@protect

```

228 \def\protected@edef{%
229     \let\@@protect\protect
230     \let\protect\@unexpandable@protect
231     \afterassignment\restore@protect
232     \edef
233 }
234 \def\protected@xdef{%
235     \let\@@protect\protect
236     \let\protect\@unexpandable@protect
237     \afterassignment\restore@protect
238     \xdef
239 }
240 \def\unrestored@protected@xdef{%
241     \let\protect\@unexpandable@protect
242     \xdef
243 }
244 \def\restore@protect{\let\protect\@@protect}

```

```

\protect The normal meaning of \protect
245 \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.

246 </2ekernel>
247 <latexrelease>\IncludeInRelease{2019/10/01}{\MakeRobust}{\MakeRobust}%
248 <*2ekernel | latexrelease>
249 \def\MakeRobust#1{%
250   \@ifundefined{\expandafter@gobble\string#1}{%
251     \@latex@error{The control sequence '\string#1' is undefined!%
252       \MessageBreak There is nothing here to make robust}%
253     \@eha
254   }%

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.

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

```

```

282 <latexrelease> {%
283 <latexrelease>   \@ifundefined{\expandafter@gobble\string#1\space}%
284 <latexrelease>   {%
285 <latexrelease>     \expandafter\let\csname
286 <latexrelease>     \expandafter@gobble\string#1\space\endcsname=#1%
287 <latexrelease>     \edef\reserved@a{\string#1}%
288 <latexrelease>     \def\reserved@b{#1}%
289 <latexrelease>     \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
290 <latexrelease>     \edef#1{%
291 <latexrelease>       \ifx\reserved@a\reserved@b
292 <latexrelease>         \noexpand\x@protect\noexpand#1%
293 <latexrelease>       \fi
294 <latexrelease>       \noexpand\protect\expandafter\noexpand
295 <latexrelease>       \csname\expandafter@gobble\string#1\space\endcsname}%
296 <latexrelease>     }%
297 <latexrelease>     {\@latex@info{The control sequence ‘\string#1’ is already robust}}%
298 <latexrelease>   }%
299 <latexrelease>}%
300 <latexrelease>\EndIncludeInRelease
301 <latexrelease>\IncludeInRelease{0000/00/00}{\MakeRobust}{\MakeRobust}%
302 <latexrelease>\let\MakeRobust\@undefined
303 <latexrelease>\EndIncludeInRelease
304 <*2ekernel>

```

`\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.

```

305 </2ekernel>
306 <*2ekernel | latexrelease>
307 <latexrelease> \IncludeInRelease{0000/00/00}%
308 <latexrelease>     {\kernel@make@fragile}{Undo robustness}%
309 \def\kernel@make@fragile#1{%
310   \@ifundefined{\expandafter@gobble\string#1\space}%
311     {}%
312     Otherwise copy \foo_ back to \foo and then undefine the payload command.
313     {%
314       \global\expandafter\let\expandafter #1\csname
315       \expandafter@gobble\string#1\space\endcsname
316       \global\expandafter\let\csname
317       \expandafter@gobble\string#1\space\endcsname\@undefined
318     }
319 <latexrelease> \EndIncludeInRelease
320 </2ekernel | latexrelease>
321 <*2ekernel>

```

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,

```
322 </2ekernel>
323 <latexrelease>\IncludeInRelease{2018-04-01}{\@ifundefined}
324 <latexrelease>\Leave commands undefined in \@ifundefined}%
325 <*2ekernel | latexrelease>

Version using \ifcsname to avoid defining undefined tokens to \relax. Defined
here to simplify using unmatched \fi.
326 \def\@ifundefined#1{%
327   \ifcsname#1\endcsname\@ifundefin@d@i\else\@ifundefin@d@ii\fi{#1}}

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

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

Now test of engine.
334 \ifx\numexpr\@undefined

Classic version (should not be needed as etex is assumed).
335 \def\@ifundefined#1{%
336   \expandafter\ifx\csname#1\endcsname\relax
337     \expandafter\@firstoftwo
338   \else
339     \expandafter\@secondoftwo
340   \fi}
341 \else\ifx\directlua\@undefined

Use the \ifcsname defined above.
342 \else

Optimised version for LuaTeX, using \lastnamedcs
343 \def\@ifundefined#1{%
344   \ifcsname#1\endcsname
345     \expandafter\ifx\lastnamedcs\relax\else\@ifundefin@d@i\fi
346   \fi
347   \@firstoftwo}

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

349 \fi
350 \fi
351 </2ekernel | latexrelease>
352 <latexrelease>\EndIncludeInRelease
353 <latexrelease>\IncludeInRelease{0000-00-00}{\@ifundefined}
354 <latexrelease>\Leave commands undefined in \@ifundefined}%
355 <latexrelease>\def\@ifundefined#1{%
356 <latexrelease>   \expandafter\ifx\csname#1\endcsname\relax
357 <latexrelease>     \expandafter\@firstoftwo
358 <latexrelease>   \else
359 <latexrelease>     \expandafter\@secondoftwo
360 <latexrelease>   \fi}
361 <latexrelease>\EndIncludeInRelease
362 <*2ekernel>
```

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

```
363 \edef\@qend{\expandafter\@cdr\string\end\@nil}
364 \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.

```
365 \long\def\@ifnextchar#1#2#3{%
366   \let\reserved@d=#1%
367   \def\reserved@a{#2}%
368   \def\reserved@b{#3}%
369   \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.

```
370 \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`.

```
371 \def\@ifnch{%
372   \ifx\@let@token\@sptoken
373     \let\reserved@c\@xifnch
374   \else
375     \ifx\@let@token\reserved@d
376       \let\reserved@c\reserved@a
377     \else
378       \let\reserved@c\reserved@b
379     \fi
380   \fi
381   \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.

```
382 \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.

```
383 \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`.

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

```

\@dblarg
\@xdblarg 385 \long\def\@dblarg#1{\kernel@ifnextchar[#{1}]{\@xdblarg{#1}}
386 \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.

387 \def\@sanitize{\@makeother\ \@makeother\\\@makeother\$\@makeother\&%
388 \@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.

389 \def \@onelevel@sanitize #1{%
390 \edef #1{\expandafter\strip@prefix
391 \meaning #1}%
392 }

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

```

12 Discretionary Hyphenation

```

\@dischyp
\@dischyp Moved here to be after the definition of \DeclareRobustCommand.
The primitive \@dischyp 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 \@dischyp by

\def\@dischyp{-}{-}

```

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

```

the \@dischyp 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 TEX 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.

```

395 \</2kernel)
396 (latexrelease)\IncludeInRelease{2017/04/15}{\@dischyp}{Use \hyphenchar in \@dischyp}

```

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

```
397 (*2ekernel)
398 \def\@latex@info#1{}
399 </2ekernel)

400 (*2ekernel | latexrelease)
401 \DeclareRobustCommand{\-}{%
402   \discretionary{%
403     \char \ifnum\hyphenchar\font<\z@
404       \defaultshyphenchar
405     \else
406       \hyphenchar\font
407     \fi
408     }{}{}}%
409 }
410 \let\@dischyph=\-
411 </2ekernel | latexrelease)
412 (latexrelease)\EndIncludeInRelease
413 (latexrelease)\IncludeInRelease{0000/00/00}{\-}{Use \hyphenchar in \-}%
414 (latexrelease)\def\-\{\discretionary{-}{}{}}
415 (latexrelease)\let\@dischyph=\-
416 (latexrelease)\EndIncludeInRelease
417 (*2ekernel)

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

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

```
446 \latexrelease\kernel@make@fragile\bigbreak
447 \latexrelease\kernel@make@fragile\break
448 \latexrelease\kernel@make@fragile\dotfill
449 \latexrelease\kernel@make@fragile\frenchspacing
450 \latexrelease\kernel@make@fragile\goodbreak
451 \latexrelease\kernel@make@fragile\hrulefill
452 \latexrelease\kernel@make@fragile\medbreak
453 \latexrelease\kernel@make@fragile\nobreak
454 \latexrelease\kernel@make@fragile\nonfrenchspacing
455 \latexrelease\kernel@make@fragile\obeylines
456 \latexrelease\kernel@make@fragile\obeyspaces
457 \latexrelease\kernel@make@fragile\slash
458 \latexrelease\kernel@make@fragile\smallbreak
459 \latexrelease\kernel@make@fragile\strut
460 \latexrelease\kernel@make@fragile\underbar
461 \latexrelease\
462 \latexrelease\EndIncludeInRelease
463 \*2kernel\
464 \2kernel\
```

File e

lalloc.dtx

13 Counters

This section deals with counter and other variable allocation.

1 (*2kernel)

The following are from plain T_EX:

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

\@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 L^AT_EX kernel commands.
16 `\newtoks\@temptokena`

`\@flushglue` Glue used for `\right-` & `\leftskip = 0pt plus 1fil`
17 `\newskip\@flushglue \@flushglue = 0pt plus 1fil`
18 `\</2kernel`

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 L^AT_EX 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 sequence, called as `\@removeelement{<element>}{<list>}{<cs>}`. Due to the implementation 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 (*2kernel)
```

The '2kernel' 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 \sim 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{\sim 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{\sim J#1}%
47 \def~{\errmessage{%
48 #2.\sim J\sim J%
49 #3\sim 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>	<p>These commands are intended for use by package and class writers, to give information to authors. The syntax is:</p> <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}

```

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

```

196 \gdef\@nodocument{%
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

`\nopcodebreak` [*i*] : *i* = 0,...,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` [*i*] : same as except negatives of its penalty
`\linebreak` [*i*] : analog of the above
`\nolinebreak` [*i*] : 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`
`\` [*length*] : initially defined to be `\vspace{<length>}\newline`
Note: `\`* adds a `\adjust{\penalty 10000}`
OBSOLETE COMMANDS (which never made it into the manual):
`\obeycr` : defines `¡CRi == \\relax`
`\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     \@tempkipa\lastskip
45     \unskip
46     \penalty #1\@getpen{#2}%
47     \ifdim\@tempkipa>\z@
48       \hskip\@tempkipa
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\@savsf
186 <latexrelease> \ifdim\@savsk>\z@
187 <latexrelease> \ignoretrue
188 <latexrelease> \ignorespaces
189 <latexrelease> \fi
190 <latexrelease> \fi}%
191 <latexrelease>\EndIncludeInRelease
192 (*2ekernel)

```

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

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

```

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

```

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      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 </2ekernel>
221 <latexrelease>\IncludeInRelease{2015/01/01}%
222 <latexrelease>         {\addpenalty}{\addpenalty}%
223 <*2ekernel | latexrelease>

```

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

```

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         \@tempkipb\lastskip

```

We have to make sure the final `\vskip` seen by `TEX` is the correct one, namely `\@tempkipb`. 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     \@tempkipa\@tempkipb
236     \advance \@tempkipb
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 -\@tempkipb
241     \penalty#1%
242     \ifdim\@tempkipa=\@tempkipb

```

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

```

243     \else

```

Combine the `prevdepth` adjustment into a single skip.

```

244     \advance\@tempkipb -\@tempkipa
245     \vskip \@tempkipb
246     \fi

```

The final skip is always the specified length.

```

247     \vskip \@tempkipa
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\cename nobreakspace \endcename

```

`\,` 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 `TEX` 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 = 0pt 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 TEX logo, adjusted so that a full stop after the logo counts as ending a
sentence.
1 <*/2kernel)
2 \DeclareRobustCommand\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
```

```
\LaTeX The LATEX 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 LATEX 2ε logo as proposed by A-W designers.
13 \DeclareRobustCommand{\LaTeXe}{\mbox{\m@th
14   \if b\expandafter\@car\fontseries\@nil\boldmath\fi
15   \LaTeX\kern.15em2$_{\textstyle\varepsilon}$}}
16 </2kernel)
```

File k

lfiles.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>\@filesw</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>\@filesw</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 \LaTeX 2_ϵ , 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> .

Historical \LaTeX 2.09 comments (not necessarily accurate any more):

```
1 (*2kernel)
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>@filesw</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@b == 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 \kernel
            12 (latexrelease)\IncludeInRelease{2017/04/15}%
            13 (latexrelease) {\document}{Save language for hyphenation}%
            14 (*2kernel | 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 fonts 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 `\normalsfcode` 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\normalsfcode\@empty
43   \ifnum\sfcode'\.=\@m
44     \let\normalsfcode\frenchspacing
45   \else
46     \let\normalsfcode\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@currsize\@empty
98 <latexrelease> \normalsize
99 <latexrelease> \everypar{}%
100 <latexrelease> \ifx\normalsfcodes\@empty
101 <latexrelease>   \ifnum\sfcode'\.=\@m
102 <latexrelease>     \let\normalsfcodes\frenchspacing
103 <latexrelease>   \else
104 <latexrelease>     \let\normalsfcodes\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 (*2kernel)

125 \@onlypreamble\document

```

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

```
126 \let\normalsfcodes\@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) \latex@error{\string\include\space cannot be nested}\@eha
174 (latexrelease) \else \@include#1 \fi}
175 (latexrelease)
176 (latexrelease)\EndIncludeInRelease
177 (*2kernel)

\@include
178 \def\@include#1 {%
179 \clearpage
180 \if@filesw
181 \immediate\write\@mainaux{\string\@input{#1.aux}}}%
182 \fi
183 \@tempswatruue
184 \if@partsw
185 \@tempswafalse
186 \edef\reserved@b{#1}%
187 \@for\reserved@a:=\@partlist\do
188 {\ifx\reserved@a\reserved@b\@tempswatruue\fi}%
189 \fi
190 \if@tempswa
191 \let\@auxout\@partaux
192 \if@filesw
193 \immediate\openout\@partaux #1.aux
194 \immediate\write\@partaux{\relax}%
195 \fi
196 \input@{#1.tex}%
197 \clearpage
198 \@writeckpt{#1}%
199 \if@filesw
200 \immediate\closeout\@partaux
201 \fi
202 \else
If the file is not included, reset \deadcycles, so that a long list of non-included
files does not generate an ‘Output loop’ error.
203 \deadcycles\z@
204 \@nameuse{cp@#1}%
205 \fi
206 \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` `a b c` -> `"a b c"`
`\unquote@name` `"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 </2kernel | 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 (*2ekernel)

```

`\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 (/2ekernel)
311 (latexrelease)\IncludeInRelease{2019/10/01}%
312 (latexrelease) {\InputIfFileExists}{Don't lose the file name}%
313 (*2ekernel | 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 (/2ekernel | 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 (*2ekernel)

```

`\input` Input a file: if the argument is given in braces use safe input macros, otherwise use T_EX’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 </2ekernel>
334 (*2ekernel | 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 </2ekernel | 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 (*2ekernel)

\@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 `intex`. 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 flacky one and make the default definition point to the new encoding. In such a case defining

```
\DeclareTextSymbol{\textdollar}{TS1}{36}
\DeclareTextSymbolDefault{\textdollar}{TS1}
```

is not enough since if typesetting in OT1 L^AT_EX 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.
2ekernel	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!
```

If you say:

```
\DeclareTextCommand
\ProvideTextCommand
\DeclareTextSymbol
   \@dec@text@cmd
   \chardef@text@cmd
   \@changed@cmd
   \@changed@x
\TextSymbolUnavailable
   \@inmathwarn
   \DeclareTextCommand{\foo}{T1}...
```

then `\foo` is defined to be `\T1-cmd \foo \T1\foo`, where `\T1\foo` is *one* control sequence, not two! We then call `\newcommand` to define `\T1\foo`.

```
3 \def\DeclareTextCommand{%
4   \@dec@text@cmd\newcommand}
5 \def\ProvideTextCommand{%
6   \@dec@text@cmd\providecommand}
7 \def\@dec@text@cmd#1#2#3{%
8   \expandafter\def\expandafter#2%
9     \expandafter{%
10      \csname#3-cmd\expandafter\endcsname
11      \expandafter#2%
12      \csname#3\string#2\endcsname
13    }%
14   \let\@ifdefinable\@rc@ifdefinable
15   \expandafter#1\csname#3\string#2\endcsname}
```

This command was introduced to fix a major bug in `\@dec@text@cmd` without changing that command itself. This was thought to be necessary because it is defined in more than one package. (Perhaps the more serious bug is to put complex low-level commands like this in packages?)

The problem it solves is that whereas both `\newcommand` and `\providecommand` (used just above) both handle the resetting of `\@ifdefinable` (following its disabling in `\@dec@text@cmd`), the primitive `\chardef` neither needs the disabling, nor does the resetting.

```
16 \def\chardef@text@cmd{%
17   \let\@ifdefinable\@ifdefinable
18   \chardef
19 }
20 \def\DeclareTextSymbol#1#2#3{%
21   \@dec@text@cmd\chardef@text@cmd#1{#2}#3\relax
22 }
```

The declarations are only available before `\begin{document}`.

```
23 \@onlypreamble\DeclareTextCommand
24 \@onlypreamble\DeclareTextSymbol
```

The sneaky bit in all this is what `\T1-cmd \foo \T1\foo` does. There are five possibilities, depending on the current values of `\protect`, `\cf@encoding` and `\ifmmode`:

- If `\protect` is `\@typeset@protect` and `\cf@encoding` is T1, then we execute `\T1\foo`. This should be the normal behaviour, and is optimized for speed.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) OT1, and `\OT1\foo` is defined, then we execute `\OT1\foo`.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) OT1, we're in text mode, and `\OT1\foo` is undefined, then we define `\OT1\foo` to be the default value of `\foo`, and execute `\OT1\foo`.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) OT1, we're in math mode, and `\OT1\foo` is undefined, then we execute the default value of `\foo`. (This is necessary so that things like `X_\copyright` work properly.)
- If `\protect` is not `\@typeset@protect` then we execute `\noexpand\foo`. For example, if we are writing to a file, then this results in `\foo` being written. If we are in a `\mark`, then `\foo` will be put in the mark—since `\foo` is robust, it will then survive all the things which may happen to it whilst it's a `\mark`.

So after all that, we will either execute the appropriate definition of `\foo` for the current encoding, or we will execute `\noexpand\foo`.

The default value of `\foo` is `\?\foo` if it is defined, and an error message otherwise.

When the encoding is changed from T1 to OT1, `\T1-cmd` is defined to be `\@changed@cmd` and `\OT1-cmd` is defined to be `\@current@cmd`. This means that the test for what the current encoding is can be performed quickly.

```

25 \def\@current@cmd#1{%
26   \ifx\protect\@typeset@protect
27     \@inmathwarn#1%
28   \else
29     \noexpand#1\expandafter\@gobble
30   \fi}

31 \def\@changed@cmd#1#2{%
32   \ifx\protect\@typeset@protect
33     \@inmathwarn#1%
34     \expandafter\ifx\csname\cf@encoding\string#1\endcsname\relax
35       \expandafter\ifx\csname ?\string#1\endcsname\relax
36         \expandafter\def\csname ?\string#1\endcsname{%
37           \TextSymbolUnavailable#1%
38         }%
39     \fi
40     \global\expandafter\let
41       \csname\cf@encoding\string#1\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 } \eoh}

```

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}
```

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
\@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 \endkernel)
77 \latexrelease)\IncludeInRelease{2017/04/15}{\DeclareTextCompositeCommand}
78 \latexrelease) {test for undeclared accent}%
79 \endkernel | 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 </2ekernel | 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 <latexrelease>     \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 <*2ekernel>
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 /<2kernel>
147 (*2kernel | latexrelease)
148 <latexrelease> \IncludeInRelease{2019/10/01}%
149 <latexrelease> { \UseTextAccent } { Make commands robust } %
```

`\UseTextAccent`
`\UseTextSymbol`
`\@use@text@encoding`

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

For a detailed discussion of this reimplementation 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 </2ekernel | latexrelease>
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{0000/00/00}%
170 <latexrelease> \UseTextAccent}{Make commands robust}%
171 <latexrelease>
172 <latexrelease>\kernel@make@fragile\UseTextAccent
173 <latexrelease>\kernel@make@fragile\UseTextSymbol
174 <latexrelease>
175 <latexrelease>\EndIncludeInRelease
176 <*2ekernel>

```

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\font@encoding{#1}%
179 \xdef\font@name{%
180 \csname\curr@fontshape/\font@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 205 %\let\@@patterns\patterns
\@@hyphenation 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 Miscellanea

`\a` The `\a` command is used to access the accent commands even when they have been redefined (for example by the `tabbing` environment). Its internal name is `\@tabacckludge`.

The `\string` within the `\csname` guards against something like `'` being active at the point of use.

```

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}{\~{}}
287 \DeclareTextCommandDefault{\textasciitilde}{\~{}}
288 \DeclareTextCommandDefault{\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}

```

We use `\protected not \DeclareRobustCommand` so that `\bigl\{` etc. works inside `\protected@edef`.

```

308 \protected\def\lbrace{\ifmmode\lbrace\else\textbraceleft\fi}
309 \protected\def\rbrace{\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   \ifmode\mathellipsis\else\textellipsis\fi}
322 \let\ldots\dots

Default definition of the commabelow accent.
323 \kernel
324 (latexrelease)\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%
325 (*2kernel | latexrelease)
326 \DeclareTextCommandDefault\textcommabelow[1]
327   {\hmode\bgroup\oalign{\null#1\crr\hidewidth\raise-.31ex
328     \hbox{\check@mathfonts\fontsize\ssf@size\z@
329       \math@fontsfalse\selectfont,}\hidewidth}\egroup}
330 (latexrelease)\EndIncludeInRelease
331 \kernel | 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 (*2kernel | latexrelease)
355 \DeclareTextCommandDefault\textcommaabove[1]{%
356   \hmode\bgroup
357   \oalign{%
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 \kernel | latexrelease)
369 (latexrelease)\IncludeInRelease{0000/00/00}{\textcommaabove}{comma above}%

```

```

370 (latexrelease)\let\textcommaabove\@undefined
371 (latexrelease)\expandafter
372 (latexrelease) \let\csmname\string\OT1\string\c-g\endcsmname\@undefined
373 (latexrelease)\expandafter
374 (latexrelease) \let\csmname\string\T1\string\c-g\endcsmname\@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\crrc\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@\crrc\hidewidth\char24\hidewidth}}\fi}
395 \DeclareTextCommand{\d}{OT1}[1]
396   {\hmode@bgroup
397     \o@lign{\relax#1\crrc\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{\texthyphenchar}{OT1}{'\-}
414 %\DeclareTextSymbol{\texthyphen}{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 \ooalign 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\crrc\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{\ooalign{\unhbox\z@\crrc
478       \hidewidth\char11\hidewidth}}\fi}
479 \DeclareTextCommand{\d}{T1}[1]
480   {\hmode@bgroup
481     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
482 \DeclareTextCommand{\k}{T1}[1]
483   {\hmode@bgroup\ooalign{\null#1\crrc\hidewidth\char12}\egroup}
484 \DeclareTextCommand{\textogonekcentered}{T1}[1]
485   {\hmode@bgroup\ooalign{%
486     \null#1\crrc\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     \oalign{%
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     \oalign{%
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{\texthyphenchar}{T1}{127}
568 %\DeclareTextSymbol{\texthyphen}{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}{0}{\textogonekcentered{0}}

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 \LaTeX 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 \oalign{%
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 (*OML)
732 \DeclareFontEncoding{OML}{}{}

```

Declare the symbols.

```

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 (*OT4)
738 \DeclareFontEncoding{OT4}{}{}
739 \DeclareFontSubstitution{OT4}{cmr}{m}{n}

```

Declare the accents.

```

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\crrc\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{\oalign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}}\fi}
759 \DeclareTextCommand{\d}{OT4}[1]
760   {\hmode@bgroup
761     \o@lign{\relax#1\crrc\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}{0}{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\cr\hidewidth\char11\hidewidth}\egroup}
829 \DeclareTextCommand{\capitalogonek}{TS1}[1]
830 {\hmode@bgroup

```

```
831 \oalign{\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 assymmetric 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{\texttildelow}{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{\textasciiacute}{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{\textasciiacute} \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{\texttimes}	\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{\textacutedbl}	\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{\textquotelleft}	\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{\textservicemark} \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\fontsize 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\crr\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{ˆ}	{e}{"00EA}
1236	\DeclareUnicodeComposite{“}	{e}{"00EB}
1237	\DeclareUnicodeComposite{‘}	\i {"00EC}
1238	\DeclareUnicodeComposite{’}	{i}{"00EC}
1239	\DeclareUnicodeComposite{ˆ}	\i {"00ED}
1240	\DeclareUnicodeComposite{’}	{i}{"00ED}
1241	\DeclareUnicodeComposite{ˆ}	\i {"00EE}
1242	\DeclareUnicodeComposite{ˆ}	{i}{"00EE}
1243	\DeclareUnicodeComposite{“}	\i {"00EF}
1244	\DeclareUnicodeComposite{“}	{i}{"00EF}
1245	\DeclareUnicodeComposite{˜}	{n}{"00F1}
1246	\DeclareUnicodeComposite{’}	{o}{"00F2}
1247	\DeclareUnicodeComposite{’}	{o}{"00F3}
1248	\DeclareUnicodeComposite{ˆ}	{o}{"00F4}
1249	\DeclareUnicodeComposite{˜}	{o}{"00F5}
1250	\DeclareUnicodeComposite{“}	{o}{"00F6}
1251	\DeclareUnicodeComposite{’}	{u}{"00F9}
1252	\DeclareUnicodeComposite{’}	{u}{"00FA}
1253	\DeclareUnicodeComposite{ˆ}	{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{ˆ}	{C}{"0108}
1266	\DeclareUnicodeComposite{ˆ}	{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{ˆ}	{G}{"011C}
1284	\DeclareUnicodeComposite{ˆ}	{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 `FOO`, the package looks to see if the encoding `FOO` has already been declared. If it has not, the file `fooenc.def` is loaded. The default encoding is set to be `FOO`.

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 wanted 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\cyrcc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrds
1447 \CYRCHRDS\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
1448 \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
1449 \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
1450 \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCCHCRS
1451 \cyrgchcrs\CYRGCHCRS\cyrgkh\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
1452 \cyrhds\CYRHDS\cyrhcrs\CYRHCRS\cyrhkh\CYRHHK\cyrhdsn
1453 \CYRHDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
1454 \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
1455 \cyrkbeak\CYRKBEAK\cyrkds\CYRKDSC\cyrkchcrs\CYRKCHCRS\cyrkhk
1456 \CYRKHK\cyrkvcrs\CYRKVCRS\cyrll\CYRL\cyrldsc\CYRLDSC\cyrllhk
1457 \CYRLHK\cyrllje\CYRLJE\cyrm\CYRM\cyrmdsc\CYRMDSC\cyrmhk\CYRMHK
1458 \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnkh\CYRNHK\cyrnje
1459 \CYRNJE\cyrnkh\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
1460 \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
1461 \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
1462 \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
1463 \cyrsftsn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
1464 \cyrt\CYRT\cyrtsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
1465 \cyru\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyry\CYRY
1466 \cyrya\CYRYA\cyryat\CYRYAT\cyrychcrs\CYRYCHCRS\cyryi\CYRYI\cyryo
1467 \CYRYO\cyryu\CYRYU\cyryz\CYRYZ\cyryzsc\CYRYZSC\cyryzh\CYRYZH
1468 \cyryzhdsc\CYRYZHDSC}%
1469 \let\update@uclc@with@cyrillic\relax
1470 }
```

Here we process each option:

```
1471 \DeclareOption*{%
1472 \let\encodingdefault\CurrentOption
```

From 2020/02/02 release onward we only load the encoding files if they haven't be loaded already. To check this we look if `\T@encoding` is already defined. If not we load (indicated by setting the switch `@tempwa` to true and we always load if we run in an older format (or rather in a rollback situation).

```

1473 \@tempswafalse
1474 \@ifl@t@r\fmtversion{2020/02/02}%
1475     {\expandafter\ifx\csname T@\CurrentOption\endcsname\relax
1476       \@tempswatrue\fi}%
1477     {\@tempswatrue}%

Load if necessary:
1478 \if@tempswa

1479 \edef\reserved@f{%
1480   \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
1481 \reserved@f
1482 \InputIfFileExists\reserved@f
1483   {\PackageError{fontenc}%
1484     {Encoding file ‘\reserved@f’ not found.%
1485     \MessageBreak
1486     You might have misspelt the name of the encoding}%
1487   {Necessary code for this encoding was not
1488     loaded.\MessageBreak
1489     Thus calling the encoding later on will
1490     produce further error messages.}}%
1491 \let\reserved@f\relax

```

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

```

1492 \expandafter\in@\expandafter{\CurrentOption}%
1493                               {T2A,T2B,T2C,X2,LCY,OT2}%
1494 \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.

```

1495 \expandafter\in@\expandafter\cyrillic\expandafter
1496                               {\@uclclist}%
1497 \ifin@
1498 \else
1499   \update@uclc@with@cyrillic
1500 \fi
1501 \fi
1502 \fi
1503 }

```

```
1504 \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 `\fontshape`, 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 \LaTeX previously used:

```
1505 %\fontencoding\encodingdefault\selectfont
```

So instead we do this here:

```
1506 \usefont\encodingdefault\familydefault\seriesdefault\shapedefault
```

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

```
1507 \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.

```
1508 \let\@elt\relax
1509 \xdef\@fontenc@load@list{\@fontenc@load@list
1510   \@elt{\csname opt@fontenc.sty\endcsname}}

1511 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
1512 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
1513 \global\let\@ifl@ter@@\@ifl@ter
1514 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
1515 </package>
```

File m

ltxcounts.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\endcsname\expandafter\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 </2kernel | 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 \TeX : 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 \TeX , which ruins any kerning between the preceding characters and whatever awaits typesetting. If you use $e\TeX$ as engine for $L^A\TeX$ (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 \TeX , 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 \LaTeX 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 eTeX 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 eTeX 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 eTeX 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 \langle latexrelease \rangle \IncludeInRelease{2015/01/01}%
6 \langle latexrelease \rangle \setlength}{Using \setlength with \dimen0}%
7 (*2ekernel | latexrelease)

8 \def\setlength#1#2{#1 #2\relax}
9 (/2ekernel | latexrelease)
10 \langle latexrelease \rangle \EndIncludeInRelease
11 \langle latexrelease \rangle \IncludeInRelease{0000/00/00}%
12 \langle latexrelease \rangle \setlength}{Using \setlength with \dimen0}%
13 \langle latexrelease \rangle \def\setlength#1#2{#1#2\relax}
14 \langle latexrelease \rangle \EndIncludeInRelease
15 (*2ekernel)

\addtolength \relax added 24 Mar 86
16 \def\addtolength#1#2{\advance#1 #2\relax}

\settoheight The obvious analogs of \settowidth.
\settodepth
\settowidth
\@settodim Clear the memory afterwards (which might be a lot).
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

ltfssbas.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 \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 `\mathgroup` introduction).²

```
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)

```
20 \expandafter\endgroup
21 \DeclareFontShape@}
```

`\DeclareFontShape@`

```
22 </2ekernel>
23 <*2ekernel | latexrelease>
24 (latexrelease)\IncludeInRelease{2020/02/02}%
25 (latexrelease) {\DeclareFontShape@}{Maybe drop one m}%
26 \def\DeclareFontShape@#1#2#3#4#5#6{%
27 \expandafter\ifx\csname #1+#2\endcsname\relax
28 \latex@error{Font family ‘#1+#2’ unknown}\@eha
29 \else
```

If the series value is incorrectly specified with an extra “m”, e.g., “mc” instead of just “c”, drop the surplus “m” but keep the “m” if it is by its own. In that case also issue a warning that the declaration needs correction.

For this we compare the given value #3 with one where we may have dropped an “m”. If nothing has changes, fine. Otherwise there was a wrong value which is now corrected in `\reservedb` so we use that and also issue a warning.

```
30 \edef\reserved@a{#3}%
31 \expandafter\series@maybe@drop@one@m\expandafter{#3}\reserved@b
32 \ifx\reserved@a\reserved@b\else
33 \latex@warning{Font shape declaration has incorrect series
34 value ‘#3’.\MessageBreak It should not contain an ‘m’!
35 Please correct it.\MessageBreak Found}%
36 \fi
37 \expandafter
38 \xdef\csname#1/#2/\reserved@b/#4\endcsname
39 {\expandafter\noexpand\csname #5\endcsname}%
40 %
```

Most of the time #6 is empty so using `\let` to `\@empty` saves on space compared to using `\def`. That’s really one of the old space saving techniques and probably not necessary these days.

```
41 \def\reserved@a{#6}%
42 \global
43 \expandafter\let\csname#5\endcsname
44 \ifx\reserved@a\@empty
45 \@empty
46 \else
47 \reserved@a
48 \fi
49 \fi
50 }
```

```

51 </2ekernel | latexrelease>
52 <latexrelease>\EndIncludeInRelease
53 <latexrelease>\IncludeInRelease{0000/00/00}%
54 <latexrelease>          {\DeclareFontShape@}{Maybe drop one m}%
55 <latexrelease>
56 <latexrelease>\def\DeclareFontShape@#1#2#3#4#5#6{%
57 <latexrelease>  \expandafter\ifx\csname #1+#2\endcsname\relax
58 <latexrelease>    \@latex@error{Font family ‘#1+#2’ unknown}\@eha
59 <latexrelease>  \else
60 <latexrelease>    \expandafter
61 <latexrelease>      \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
62 <latexrelease>                                                \csname #5\endcsname}%
63 <latexrelease>    \def\reserved@a{#6}%
64 <latexrelease>    \global
65 <latexrelease>    \expandafter\let\csname#5\endcsname\expandafter\endcsname
66 <latexrelease>    \ifx\reserved@a\@empty
67 <latexrelease>      \@empty
68 <latexrelease>    \else
69 <latexrelease>      \reserved@a
70 <latexrelease>    \fi
71 <latexrelease>  \fi
72 <latexrelease> }
73 <latexrelease>\EndIncludeInRelease
74 <*/2ekernel)

```

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

```

75 \def\DeclareFixedFont#1#2#3#4#5#6{%
76   \begingroup
77     \math@fontsfalse
78     \every@math@size{}%
79     \fontsize{#6}\z@
80     \usefont{#2}{#3}{#4}{#5}%
81     \global\expandafter\let\expandafter#1\the\font
82   \endgroup
83 }

```

`\do@subst@correction`

```

84 \def\do@subst@correction{%
85   \xdef\subst@correction{%
86     \font@name
87     \global\expandafter\font
88     \csname \curr@fontshape/\f@size\endcsname
89     \noexpand\fontname\font
90     \relax}%

```

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

```

91   \aftergroup\subst@correction
92 }

```

`\DeclareFontFamily`

```

93 \def\DeclareFontFamily#1#2#3{%

```

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

```

94 % \@tempwafalse
95 % \def\reserved@b{#1}%
96 % \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
97 %     \ifx\reserved@b\reserved@c \@tempwatrue\fi}%
98 % \cdp@list
99 % \if@tempswa
100 \@ifundefined{T@#1}%
101   {%
102     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
103   }%
104   {%

```

Now we have to define the macro $\langle\#1\rangle+\langle\#2\rangle$ 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`.

```
105     \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`.

```

106     \global
107     \expandafter\let\csname #1+#2\expandafter\endcsname
108         \ifx \reserved@a\@empty
109             \@empty
110         \else \reserved@a
111         \fi
112     }%
113 }

```

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

```

114 \let\cdp@list\@empty
115 \@onlypreamble\cdp@list

```

`\cdp@elt`

```

116 \let\cdp@elt\relax
117 \@onlypreamble\cdp@elt

```

`\DeclareFontEncoding`

```
118 \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.

```

119     \begingroup
120     \nfss@catcodes
121     \expandafter\endgroup
122     \DeclareFontEncoding@}
123 \@onlypreamble\DeclareFontEncoding

```

```

124 \def\DeclareFontEncoding@#1#2#3{%
125     \expandafter
126     \ifx\csname T@#1\endcsname\relax
127         \def\cdp@elt{\noexpand\cdp@elt}%
128         \xdef\cdp@list{\cdp@list\cdp@elt{#1}%

```

```

129             {\default@family}{\default@series}%
130             {\default@shape}}}%

```

To support encoding dependent commands (like accents) we initialise the command `\<encoding>-cmd` to be `\@changed@cmd`. (See `ltoutenc.dtx` for details.)

```

131     \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
132 \else
133     \@font@info{Redefining font encoding #1}%
134 \fi
135 \global\@namedef{T@#1}{#2}%
136 \global\@namedef{M@#1}{\default@M#3}%

```

Keep a record of the last encoding being declared:

```

137 \xdef\LastDeclaredEncoding{#1}%
138 }
139 \@onlypreamble\DeclareFontEncoding@

```

`\LastDeclaredEncoding` The last encoding being declared by `\DeclareFontEncoding`.

```

140 \def\LastDeclaredEncoding{}

```

`\DeclareFontSubstitution`

```

141 \def\DeclareFontSubstitution#1#2#3#4{%
142 \expandafter
143 \ifx\csname T@#1\endcsname\relax
144 \latex@error{Encoding scheme ‘#1’ unknown}\@eha
145 \else
146 \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`.

```

147     \edef\reserved@a{#1}%
148     \toks@{}%
149     \def\cdp@elt##1##2##3##4{%
150         \def\reserved@b{##1}%
151         \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.

```

152         \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
153         \else

```

If `\reserved@a` and `\reserved@b` differ then we simply copy from the old list to the new.

```

154         \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
155         \fi}%
156     \cdp@list
157     \xdef\cdp@list{\the\toks@}%
158 \endgroup
159 \global
160 \@namedef{D@#1}{%
161     \def\default@family{#2}%

```

```

162         \def\default@series{#3}%
163         \def\default@shape{#4}%
164         }%
165     \fi
166 }
167 \@onlypreamble\DeclareFontSubstitution

```

\DeclareFontEncodingDefaults

```

168 \def\DeclareFontEncodingDefaults#1#2{%
169     \ifx\relax#1\else
170         \ifx\default@T\@empty\else
171             \@font@info{Overwriting encoding scheme text defaults}%
172             \fi
173             \gdef\default@T{#1}%
174             \fi
175         \ifx\relax#2\else
176             \ifx\default@M\@empty\else
177                 \@font@info{Overwriting encoding scheme math defaults}%
178                 \fi
179                 \gdef\default@M{#2}%
180             \fi
181 }
182 \@onlypreamble\DeclareFontEncodingDefaults

```

\default@T

\default@M

```

183 \let\default@T\@empty
184 \let\default@M\@empty

```

\DeclarePreloadSizes

```

185 \def\DeclarePreloadSizes#1#2#3#4#5{%
186     \@ifundefined{T@#1}%
187     {\@latex@error{Encoding scheme ‘#1’ unknown}\@eha}%
188     {%

```

Don't know at the moment what this group here does!

```

189     \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).

```

190     \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 *TEXbook*: 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.)

```

191         \if>##1>%
192             \let\reserved@f\relax
193         \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

³We cannot use `\@tempa` since it is needed in `\pickup@font`.

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.

```
194         \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
195         \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.

```
196         \global\expandafter\let\font@name\relax
197         \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.

```
198         \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.

```
199     \reserved@f#5,,%
200     \endgroup
201     }%
202 }
203 \@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.

```
204 \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.

```
\DeclareMathSizes*
205 \def\DeclareMathSizes{%
206   \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
207   {\@DeclareMathSizes{}}%
208 \@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}.
```

```
209 \ifx\@DeclareMathSizes\@empty\else
210 \ifx\@DeclareMathSizes\@empty\else
211 \ifx\@DeclareMathSizes\@empty\else
212 \ifx\@DeclareMathSizes\@empty\else
213 \def\@DeclareMathSizes #1#2#3#4#5{%
214   \@defaultunits\dimen@ #2pt\relax\@nnil
215   \if $#3$%
216     \expandafter\let\csname S@#1strip@pt\dimen@\endcsname\math@fontsfalse
217   \else
218     \@defaultunits\dimen@ii #3pt\relax\@nnil
219     \@defaultunits\@tempdima #4pt\relax\@nnil
220     \@defaultunits\@tempdimb #5pt\relax\@nnil
221     \toks@{#1}%
\ifx\@DeclareMathSizes\@empty\else
\ifx\@DeclareMathSizes\@empty\else
\ifx\@DeclareMathSizes\@empty\else
\ifx\@DeclareMathSizes\@empty\else
```

```

222   \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
223     \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
224     \gdef\noexpand\sfs@size{\strip@pt\@tempdima}%
225     \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
226     \the\toks@
227   }%
228 \fi
229 }%
230 </2ekernel | latexrelease)
231 (latexrelease)\EndIncludeInRelease
232 (latexrelease)\IncludeInRelease{0000/00/00}{\@DeclareMathSizes}%
233 (latexrelease)      {Arbitrary units in \DeclareMathSizes}%
234 (latexrelease)\def\@DeclareMathSizes#1#2#3#4#5{%
235 (latexrelease)      \@defaultunits\dimen@#2pt\relax\@nnil
236 (latexrelease)      \if$#3$%
237 (latexrelease)        \expandafter \let
238 (latexrelease)          \csname S@\strip@pt\dimen@\endcsname
239 (latexrelease)          \math@fontsfalse
240 (latexrelease)      \else
241 (latexrelease)        \expandafter \gdef
242 (latexrelease)          \csname S@\strip@pt\dimen@\endcsname
243 (latexrelease)          {\gdef\tf@size{#3}\gdef\sfs@size{#4}%
244 (latexrelease)          \gdef\ssf@size{#5}%
245 (latexrelease)          #1%
246 (latexrelease)          }%
247 (latexrelease)      \fi}%
248 (latexrelease)\EndIncludeInRelease
249 (*2ekernel)
250 \@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
`\f@encoding` 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.

```

251 \DeclareRobustCommand\fontencoding[1]{%
252   \expandafter\ifx\csname T@#1\endcsname\relax
253   \@latexerror{Encoding scheme ‘#1’ unknown}\@eha
254   \else
255   \edef\f@encoding{#1}%
256   \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`.

```

257     \let\enc@update\relax
258     \else

```

If current and new encoding differ we define the macro `\enc@update` to contain all updates necessary at `\selectfont` time.

```

259         \let\enc@update\@@enc@update
260         \fi
261     \fi
262 }

```

`\@@enc@update`

```
263 \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`.

```

264         \expandafter
265         \let
266         \csname\cf@encoding -cmd\endcsname
267         \@changed@cmd

```

Then we turn the one for the new encoding to `\@current@cmd` (see `ltoutenc.dtx` for further explanations).

```

268         \expandafter
269         \let
270         \csname\f@encoding-cmd\endcsname
271         \@current@cmd

```

We execute the default settings `\default@T`, followed by the one for the new encoding.

```

272         \default@T
273         \csname T@\f@encoding\endcsname

```

Finally we change the default substitution values, disable `\enc@update` and make `\f@encoding` officially the current encoding.

```

274         \csname D@\f@encoding\endcsname
275         \let\enc@update\relax
276         \let\cf@encoding\f@encoding
277 }

```

`\enc@update` The default action in `\selectfont` is to do nothing.

```
278 \let\enc@update\relax
```

`\fontfamily`

```
\f@family 279 \DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
```

`\fontseries` There are now defined later (and differently).

```
\f@series 280 %\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
```

```
\fontshape 281 %\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!

```
282 \DeclareRobustCommand\usefont[4]{\fontencoding{#1}%
```

```

283 \edef\f@family{#2}%
284 \edef\f@series{#3}%
285 \edef\f@shape{#4}\selectfont
286 \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.

```

287 \DeclareRobustCommand\linespread[1]
288   {\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.

```

289 \DeclareRobustCommand\fontsize[2]
290   {\set@fontsize\baselinestretch{#1}{#2}}

```

`\f@linespread` This macro holds the current internal value for `\baselinestretch`.

```

291 \let\f@family\@empty
292 \let\f@series\@empty
293 \let\f@shape\@empty
294 \let\f@size\@empty
295 \let\f@baselineskip\@empty
296 \let\f@linespread\@empty

```

`\cf@encoding`

```

297 \let\f@encoding\@empty
298 \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.

```
299 \def\@defaultunits{\afterassignment\remove@to@nnil}
```

`\strip@pt` This macro strips the characters pt produced by using `\the` on a dimen register.

```

\rem@pt
300 \begingroup
301   \catcode‘P=12
302   \catcode‘T=12
303   \lowercase{
304     \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
305   \expandafter\endgroup\x
306 \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.

```
307 \DeclareRobustCommand\mathversion[1]
308     {\@nomath\mathversion
309      \expandafter\ifx\csname mv@#1\endcsname\relax
310      \@latex@error{Math version '#1' is not defined}\@eha\else
311      \edef\math@version{#1}%
```

We need to force a math font setup both now and at the point where we return to the previous math version. Forcing a math font setup can simply be done by setting `\glb@currsiz` to an invalid value since this will trigger the setup when the formula starts.

```
312     \gdef\glb@currsiz{}%
```

When the scope of the current `\mathversion` ends we need to restore the old setup. However this time we need to force it directly at least if we are inside math, otherwise we could wait. Another way to enhance this code here is to do the setting only if the version really has changed after all. This might be interesting in case of `amstext` and `boldsymbol`.

```
313     \aftergroup\glb@settings
314     \fi}
```

If \TeX would support a hook just before the end of a formula (opposite of `\everymath` so to speak) the implementation of the algorithm would be much simpler because in that case we would set up the correct math fonts at this point without having to worry about incorrect settings due to nesting. The same would be true if in \LaTeX the use of $\$$ (as the primitive \TeX command) would be impossible and instead only a higher-level interface would be available. Note that this does not mean that a $\$$ couldn't be the short-hand for starting and stopping that higher-level interface, it only means that the direct \TeX function must be hidden.

Anyway, since we don't have this and won't have it in $\LaTeX 2_\epsilon$ we need to implement it in a somewhat slower way.

We test for the current math font setup on entry of a formula, i.e., on the hooks `\everymath` and `\everydisplay`. But since these hooks may contain user data we provide ourselves with an internal version of these hooks which stays frozen.

```
\frozen@everymath New internal names for \everymath and \everydisplay.
\frozen@everydisplay 315 \let\frozen@everymath\everymath
316 \let\frozen@everydisplay\everydisplay
```

```
\everymath Now we provide now user hooks that will be called in the frozen internals.
\everydisplay 317 \newtoks\everymath
318 \newtoks\everydisplay
```

```
\frozen@everymath Now we define the behaviour of the frozen hooks: first check the math setup then
call the user hook.
```

```
319 \frozen@everymath = {\check@mathfonts
320                      \the\everymath}
```

```
\frozen@everydisplay Ditto for the display hook.
```

```
321 \frozen@everydisplay = {\check@mathfonts
322                      \the\everydisplay}
```

`\curr@math@size` This holds locally the current math size.
 323 `\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.

```
324 \def\pickup@font{%
325     \expandafter \ifx \font@name \relax
326         \define@newfont
327     \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`

```
328 {\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.

```
329 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}%
330                                     \def\f@family{#2}%
331                                     \def\f@series{#3}%
332                                     \def\f@shape{#4}%
333                                     \def\f@size{#5}}}
```

`\curr@fontshape` Abbreviation which may get removed again for speed.

```
334 \def\curr@fontshape{\f@encoding/\f@family/\f@series/\f@shape}
```

`\define@newfont` Now we can tackle the problem of defining a new font.

```
335 \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.

```
336 \begingroup
337     \let\typeout\@font@info
338     \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`.

```
339     \expandafter\expandafter\expandafter
340         \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.

```

341 %   \expandafter\ifx
342 %       \csname\curr@fontshape\endcsname \relax
343 %   \try@load@fontshape % try always
344 %   \fi
345 \expandafter\ifx
346     \csname\curr@fontshape\endcsname \relax
347     \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).

```

348 %       \csname\curr@fontshape\endcsname
349 %       \extract@font\fi

```

We are nearly finished and must only restore the `\escapechar` by closing the group.

```

350 \endgroup}

```

```

351 \def\try@load@fontshape{%
352   \expandafter
353   \ifx\csname \f@encoding+\f@family\endcsname\relax
354     \@font@info{Trying to load font information for
355               \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.

```

356   \global\expandafter\let
357   \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).

```

358   \nfss@catcodes
359   \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.

```

360   \edef\reserved@a{%
361     \lowercase{%
362       \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}%
363   \reserved@a\relax
364     {\@input@\f@encoding\f@family.fd}}%
365 \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.

```
366 \def\nfss@catcodes{%
```

We start by making @ a letter and ignoring all blanks and newlines.

```
367     \makeatletter
368     \catcode'\ 9%
369     \catcode'\^^I9%
370     \catcode'\^^M9%
```

Then we set up \, {, }, # and % in case an .fd file is loaded during a verbatim environment.

```
371     \catcode'\\\z@
372     \catcode'\{\@ne
373     \catcode'\}\@tw@
374     \catcode'\#6%
375     \catcode'\^7%
376     \catcode'\%14%
```

The we make sure that the important syntax parts have the right `\catcode`.

```
377     \@makeother\<%
378     \@makeother\>%
379     \@makeother\*%
380     \@makeother\.%
381     \@makeother\-%
382     \@makeother\/%
383     \@makeother\[%
384     \@makeother\]%
385     \@makeother\'%
386     \@makeother\'%
387     \@makeother\"%
388 }
```

`\LoadFontDefinitionFile` Load and .fd files for some encoding and family (if it exists).

```
389 </2ekernel>
390 <{*2ekernel | latexrelease>
391 <latexrelease>\IncludeInRelease{2020/02/02}%
392 <latexrelease>           {\LoadFontDefinitionFile}{Loading .fd files}%
393 \def\LoadFontDefinitionFile#1#2{%
394   \begingroup
395     \edef\f@encoding{#1}%
396     \edef\f@family{#2}%
397     \try@load@fontshape
398   \endgroup
399 }
400 </2ekernel | latexrelease>
401 <latexrelease>\EndIncludeInRelease
402 <latexrelease>\IncludeInRelease{0000/00/00}%
403 <latexrelease>           {\LoadFontDefinitionFile}{Loading .fd files}%
404 <latexrelease>
405 <latexrelease>\let\LoadFontDefinitionFile\@undefined
406 <latexrelease>\EndIncludeInRelease
407 <{*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 that 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 \TeX 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).

```
408 \if2kernel)
409 (*2kernel|latexrelease)
410 (latexrelease) \IncludeInRelease{2020/02/02}%
411 (latexrelease)      {\DeclareFontFamilySubstitution}{Provide family substitution}%
412 \begingroup
413 \nfss@catcodes
414 \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.

```
415 \LoadFontDefinitionFile{#1}{#2}%
416 \LoadFontDefinitionFile{#1}{#3}%
417 \DeclareFontShape{#1}{#2}{m}{it}{<->ssub * #3/m/it}{}%
418 \DeclareFontShape{#1}{#2}{m}{n}{<->ssub * #3/m/n}{}%
419 \DeclareFontShape{#1}{#2}{m}{sc}{<->ssub * #3/m/sc}{}%
420 \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.

```
421 \DeclareFontShape{#1}{#2}{m}{sw}{<->ssub * #3/m/sw}{}%
422 \DeclareFontShape{#1}{#2}{m}{scit}{<->ssub * #3/m/scit}{}%
423 \DeclareFontShape{#1}{#2}{m}{scsl}{<->ssub * #3/m/scsl}{}%
```

Same game with b and bx, for other weights you are on your own:

```
424 \DeclareFontShape{#1}{#2}{b}{it}{<->ssub * #3/b/it}{}%
425 \DeclareFontShape{#1}{#2}{b}{n}{<->ssub * #3/b/n}{}%
426 \DeclareFontShape{#1}{#2}{b}{scit}{<->ssub * #3/b/scit}{}%
427 \DeclareFontShape{#1}{#2}{b}{scsl}{<->ssub * #3/b/scsl}{}%
428 \DeclareFontShape{#1}{#2}{b}{sc}{<->ssub * #3/b/sc}{}%
429 \DeclareFontShape{#1}{#2}{b}{sl}{<->ssub * #3/b/sl}{}%
430 \DeclareFontShape{#1}{#2}{b}{sw}{<->ssub * #3/b/sw}{}%
431 \DeclareFontShape{#1}{#2}{bx}{it}{<->ssub * #3/bx/it}{}%
432 \DeclareFontShape{#1}{#2}{bx}{n}{<->ssub * #3/bx/n}{}%
433 \DeclareFontShape{#1}{#2}{bx}{scit}{<->ssub * #3/bx/scit}{}%
434 \DeclareFontShape{#1}{#2}{bx}{scsl}{<->ssub * #3/bx/scsl}{}%
435 \DeclareFontShape{#1}{#2}{bx}{sc}{<->ssub * #3/bx/sc}{}%
436 \DeclareFontShape{#1}{#2}{bx}{sl}{<->ssub * #3/bx/sl}{}%
437 \DeclareFontShape{#1}{#2}{bx}{sw}{<->ssub * #3/bx/sw}{}%
```

```

438 }
439 \endgroup
440 </2ekernel | latexrelease>
441 (latexrelease)\EndIncludeInRelease
442 (latexrelease)\IncludeInRelease{0000/00/00}%
443 (latexrelease)      {\DeclareFontFamilySubstitution}{Provide family substitution}%
444 (latexrelease)
445 (latexrelease)\let\DeclareFontFamilySubstitution\@undefined
446 (latexrelease)\EndIncludeInRelease
447 (*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!

```

448 </2ekernel)
449 (*2ekernel | latexrelease)
450 (latexrelease)\IncludeInRelease{2019/10/01}%
451 (latexrelease)      {\DeclareErrorFont}{No side effects please}%
452 \def\DeclareErrorFont#1#2#3#4#5{%
453     \xdef\error@fontshape{%
454         \noexpand\expandafter\noexpand\split@name\noexpand\string
455         \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
456         \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.

```

457 %     \gdef\f@encoding{#1}%
458     \gdef\default@family{#2}%
459     \gdef\default@series{#3}%
460     \gdef\default@shape{#4}%
461 }
462 </2ekernel | latexrelease>
463 (latexrelease)\EndIncludeInRelease
464 (latexrelease)\IncludeInRelease{0000/00/00}%
465 (latexrelease)      {\DeclareErrorFont}{No side effects please}%
466 (latexrelease)
467 (latexrelease)\def\DeclareErrorFont#1#2#3#4#5{%
468 (latexrelease)      \xdef\error@fontshape{%
469 (latexrelease)          \noexpand\expandafter\noexpand\split@name\noexpand\string
470 (latexrelease)          \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
471 (latexrelease)          \noexpand\@nil}%
472 (latexrelease)      \gdef\default@family{#2}%
473 (latexrelease)      \gdef\default@series{#3}%
474 (latexrelease)      \gdef\default@shape{#4}%
475 (latexrelease)      \global\let\f@family\default@family
476 (latexrelease)      \global\let\f@series\default@series
477 (latexrelease)      \global\let\f@shape\default@shape
478 (latexrelease)      \gdef\f@size{#5}%
479 (latexrelease)      \gdef\f@baselineskip{#5pt}%
480 (latexrelease)}

```

```

481 (latexrelease)\EndIncludeInRelease
482 (*2ekernel)
483 \@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!

```

484 (/2ekernel)
485 (latexrelease)\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
486 (latexrelease)           {Font substitution in preamble}%
487 (*2ekernel|latexrelease)
488 \def\wrong@fontshape{%
489   \csname D@f@encoding\endcsname % install defaults if in math

```

We remember the wanted `\curr@fontshape` combination which we will need in a moment.

```

490   \edef\reserved@a{\csname\curr@fontshape\endcsname}%
491   \ifx\last@fontshape\reserved@a
492     \errmessage{Corrupted NFSS tables}%
493     \error@fontshape
494   \else

```

Then we warn the user about the mess and set the shape to its default.

```
495   \let\f@shape\default@shape
```

If the combination is not known, try the default *series*.

```

496   \expandafter\ifx\csname\curr@fontshape\endcsname\relax
497     \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!

```

498     \expandafter
499     \ifx\csname\curr@fontshape\endcsname\relax
500       \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}`.

```

501       \begingroup
502         \try@load@fontshape
503       \endgroup
504     \fi \fi
505   \fi

```

At this point a valid `\curr@fontshape` combination must have been found. We inform the user about this fact.

The `\expandafter\string` here stops T_EX 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.

```

506     \@font@warning{Font shape ‘\expandafter\string\reserved@a’
507         \expandafter@gobble\string\undefined\MessageBreak
508         using ‘\curr@fontshape’ instead@\wrong@font@char}%
509     \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.

```

510     \gdef\@defaultsubs{%
511         \@font@warning{Some font shapes were not available, defaults
512             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.

```

513     \global\expandafter\expandafter\expandafter\let
514         \expandafter\reserved@a
515         \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.

```

516     \xdef\font@name{%
517         \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.

```

518     \pickup@font}
519 </2kernel | latexrelease>
520 <latexrelease>\EndIncludeInRelease
521 <latexrelease>\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
522 <latexrelease>           {Font substitution in preamble}%
523 <latexrelease>\def\wrong@fontshape{%
524 <latexrelease>     \csname D@\f@encoding\endcsname
525 <latexrelease>     \edef\reserved@a{\csname\curr@fontshape\endcsname}%
526 <latexrelease>     \ifx\last@fontshape\reserved@a
527 <latexrelease>         \errmessage{Corrupted NFSS tables}%
528 <latexrelease>         \error@fontshape
529 <latexrelease>     \else
530 <latexrelease>         \let\f@shape\default@shape
531 <latexrelease>         \expandafter\ifx\csname\curr@fontshape\endcsname\relax
532 <latexrelease>             \let\f@series\default@series
533 <latexrelease>             \expandafter
534 <latexrelease>                 \ifx\csname\curr@fontshape\endcsname\relax
535 <latexrelease>                     \let\f@family\default@family
536 <latexrelease>                 \fi \fi
537 <latexrelease>     \fi
538 <latexrelease>     \@font@warning{Font shape
539 <latexrelease>         ‘\expandafter\string\reserved@a’

```

```

540 (latexrelease)      \expandafter\@gobble\string\@undefined
541 (latexrelease)      \MessageBreak
542 (latexrelease)      using '\curr@fontshape' instead\@wrong@font@char}%
543 (latexrelease)      \global\let\last@fontshape\reserved@a
544 (latexrelease)      \gdef\@defaultsubs{%
545 (latexrelease)        \@font@warning{Some font shapes were not available,
546 (latexrelease)          defaults substituted.\@gobbletwo}}%
547 (latexrelease)      \global\expandafter\expandafter\expandafter\let
548 (latexrelease)        \expandafter\reserved@a
549 (latexrelease)        \csname\curr@fontshape\endcsname
550 (latexrelease)      \xdef\font@name{%
551 (latexrelease)        \csname\curr@fontshape/\f@size\endcsname}%
552 (latexrelease)      \pickup@font}
553 (latexrelease)\EndIncludeInRelease
554 (*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.

```
555 \let\@wrong@font@char\@empty
```

`\@@defaultsubs` See above.

```
\@defaultsubs 556 \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).

```
557 \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.

```
558 \let\install@mathalphabet\gdef
```

`\math@fonts`

```
559 \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.

```
560 %\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.

```
561 % \begingroup
```

We set the math fonts for the *family* in question by calling `\getanddefine@fonts` in the correct environment.

```
562 % \escapechar\m@ne
```

```
563 % \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
```

We globally select the math fonts...

```
564 % \globaldefs\@ne \math@fonts
```

... and close the group to restore `\globaldefs` and `\escapechar`.

```
565 % \endgroup
```

As long as no *size* or *version* change occurs the \langle *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 *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 *math alphabet identifier* \rangle and then calling the new definition to switch to the wanted \langle *math group* \rangle . Now we define the \langle *math alphabet identifier* \rangle as a call to the `\use@mathgroup` command.

```
566 % \xdef#1{\noexpand\use@mathgroup\noexpand#2%
567 %         {\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.

```
568 % \expandafter\extract@alph@from@version
569 %     \csname mv@\math@version\expandafter\endcsname
570 %     \expandafter{\number\csname c@mv@\math@version\endcsname}%
571 %     #1%
572 %     \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.

```
573 %\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.

```
574 \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`.

```
575 \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`.

```
576 \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`.

```
577 \def\reserved@c###1{\gdef#1{##1###1##3}}%
```

Then we execute our auxiliary macro.

```
578 \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.

```
579 \def\reserved@a\select@group#3##1##2\@nil{%
```

This macro can now directly rebuild the math version definition by calling `\reserved@c`:

```
580 \reserved@c{%  
581 \getanddefine@fonts{#2}##2%  
582 \install@mathalphabet#3{%  
583 \relax\ifmmode \else \non@alpherr#3\fi  
584 \use@mathgroup##1{#2}}%
```

In addition it defines the alphabet the way it should be used from now on.

```
585 \gdef#3{\relax\ifmmode \else \non@alpherr#3\fi  
586 \use@mathgroup##1{#2}}%
```

Finally, we only have to call this macro `\reserved@a` on the old definitions recorded in `\reserved@b`:

```
587 \expandafter\reserved@a\reserved@b\@nil  
588 }
```

`\math@bgroup` Here are the default definitions for `\math@bgroup` and `\math@egroup`. We use `\bgroup` instead of `\begingroup` to avoid ‘leaking out’ of style changes. This has the side effect of always producing mathord atoms.

```
589 \let\math@bgroup\bgroup  
590 \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 `mthscale.sty`.

```

591 \gdef\calculate@math@sizes{%
592   \@font@info{Calculating\space math\space sizes\space for\space
593     size\space <\f@size>}%
594   \dimen@\f@size \p@
595   \@tempdimb \defaultscritratio \dimen@
596   \dimen@ \defaultscritscritratio \dimen@
597   \expandafter\xdef\csname S@\f@size\endcsname{%
598     \gdef\noexpand\tf@size{\f@size}%
599     \gdef\noexpand\sf@size{\strip@pt\@tempdimb}%
600     \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
601     \noexpand\math@fontstrue}}

```

`\defaultscritratio` The default ratio for math sizes is:
`\defaultscritscritratio` 1 to `\defaultscritratio` to `\defaultscritscritratio`.
 By default this is 1 to .7 to .5.

```

602 \def\defaultscritratio{.7}
603 \def\defaultscritscritratio{.5}

```

`\noaccents@` If we don't have a definition for `\noaccents@` we provide a dummy.

```

604 \ifx\noaccents@\@undefined
605   \let\noaccents@\@empty
606 \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.

```

607 </2kernel>
608 (latexrelease)\IncludeInRelease{2017/01/01}{\showhyphens}%
609 (latexrelease)           {XeTeX support for \showhyphens}%
610 (*2kernel | latexrelease)
611 \ifx\XeTeXcharclass\@undefined

```

Version for engines other than XeT_EX.

```

612 \DeclareRobustCommand\showhyphens[1]{%
613   \setbox0\vbox{%
614     \color@begingroup
615     \everypar{}%
616     \parfillskip\z@skip\hsize\maxdimen
617     \normalfont
618     \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
619     \color@endgroup}}

```

```
620 \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.

```

621 \DeclareRobustCommand\showhyphens[1]{%
622   \setbox0\vbox{%

```

```

623 \usefont{TU}{lmr}{m}{n}%
624 \hsize 1sp %
625 \hbadness\@M
626 \hfuzz\maxdimen
627 \tracingonline\z@
628 \everypar={}%
629 \leftskip\z@skip
630 \rightskip\z@skip
631 \parfillskip\z@skip
632 \hyphenpenalty=-\@M
633 \pretolerance\m@ne
634 \interlinepenalty\z@
635 \clubpenalty\z@
636 \widowpenalty\z@
637 \brokenpenalty1127 %
638 \setbox\z@\hbox{}%
639 \noindent
640 \hskip\z@skip
641 #1%
642 \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`).

```

643 \loop
644 \@tempswafalse
645 \ifnum\lastnodetype=11\unskip\@tempswatruе\fi
646 \ifnum\lastnodetype=12\unkern\@tempswatruе\fi
647 \ifnum\lastnodetype=13 %
648 \count@\lastpenalty
649 \unpenalty\@tempswatruе
650 \fi
651 \ifnum\lastnodetype=\@ne
652 \setbox\tw@\lastbox\@tempswatruе
653 \setbox0\hbox{\unhbox\tw@\unskip\unskip\unpenalty
654 \ifnum\count@=1127 \else\ \fi
655 \unhbox0}%
656 \count@\z@
657 \fi
658 \if@tempswa
659 \repeat
660 \hbadness\z@
661 \hsize\maxdimen
662 \showboxdepth\z@
663 \tolerance\m@ne
664 \hyphenpenalty\z@
665 \noindent\unhbox\z@
666 }}
667 \fi
668 </2ekernel | latexrelease>
669 <latexrelease>\EndIncludeInRelease
670 <latexrelease>\IncludeInRelease{0000/00/00}{\showhyphens}%
671 <latexrelease> {XeTeX support for \showhyphens}%
672 <latexrelease>\gdef\showhyphens#1{%

```

```

673 (latexrelease) \setbox0\vbox{%
674 (latexrelease)   \color@beingroup
675 (latexrelease)   \everypar{}%
676 (latexrelease)   \parfillskip\z@skip\hsize\maxdimen
677 (latexrelease)   \normalfont
678 (latexrelease)   \pretolerance\m@ne\tolerance\m@ne
679 (latexrelease)   \hbadness\z@\showboxdepth\z@\ #1%
680 (latexrelease)   \color@endgroup}}
681 (latexrelease)\EndIncludeInRelease
682 (*2ekernel)

\addto@hook We need a macro to add tokens to a hook.
683 \long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}

  \@vpt
684 \def\@vpt{5}

  \@vipt
685 \def\@vipt{6}

  \@viipt
686 \def\@viipt{7}

\@viiipt
687 \def\@viiipt{8}

  \@ixpt
688 \def\@ixpt{9}

  \@xpt
689 \def\@xpt{10}

  \@xipt
690 \def\@xipt{10.95}

  \@xiipt
691 \def\@xiipt{12}

  \@xivpt
692 \def\@xivpt{14.4}

  \@xviipt
693 \def\@xviipt{17.28}

  \@xxpt
694 \def\@xxpt{20.74}

  \@xxvpt
695 \def\@xxvpt{24.88}
696 (/2ekernel)

```

File p

ltfssaxes.dtx

This file contains the implementation for handling extra axes splitting the series and the values into sub-categories. 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 Font Selection Scheme.

Everything in the this file got introduced 2020/02/02, so we do a single rollback (for now).

```
1 (*2ekernel)
2 </2ekernel)
3 (*2ekernel | latexrelease)
4 (latexrelease) \IncludeInRelease{2020/02/02}%
5 (latexrelease)   {\DeclareFontSeriesChangeRule}{Series change rules}%
```

28 Changing the font series

In the original NFSS implementation the series was a single attribute stored in `\f@series` and so one always had to specify both weight and width together. This means it was impossible to typeset, a paragraph in a condensed font and inside have a few words in bold weight (but still condensed) without doing this manually by requesting `\fontseries{bc}\selectfont`.

The new implementation now works differently by looking both at the current value of `\f@series` and the requested new series and out of that combination selects a resulting series value. Thus, if the current series is `c` and we ask for `b` we now get `bc`.

This is done by consulting a simple lookup table. This table is configurable (though most likely that flexibility will seldom of ever be needed) Adding or changing entries in this table are done with `\DeclareFontSeriesChangeRule`.

28.1 The series lookup table

`\DeclareFontSeriesChangeRule` The `\DeclareFontSeriesChangeRule` defines entries in a simple database (implemented as a set of commands) that define mappings between from an existing series and requested new series and maps that to a result series (and additionally offers an alternative if the desired one is not existing):

```
#1 current \f@series
#2 requested new series
#3 result (if that exist for the given font family)
#4 alternative result (if #3 does not exist)
```

If an `.fd` file has its own substitution rules then `#3` exist and thus `#4` is not applied.

If there is no matching database entry or if neither the result nor the alternate result exist in the font family the requested new series is used (which then may trigger substitutions later on).

```
6 \def\DeclareFontSeriesChangeRule#1#2#3#4{%
7   \@namedef{series@#1@#2}{#3}{#4}}
```

28.2 Mapping rules for series changes

The rules set up use explicit series values not `\. .default` indirections; my current feeling is that this is in fact better.

With 9 weights and 9 width classes this table is getting a bit large in the end (324 entries) but on the other hand it doesn't change and accessing speed and it is fast this way.

We could alternatively split the axis and maintain weight and width separately, but that would take more processing time and would not allow for setting up explicit exceptions nicely (not sure that this would ever get used though).

Design considerations for mapping entries:

- We make `m` to reset both weight and width (as this is how it always worked). To reset just the width `?m` is provided and to reset just the weight `m?`.
- We do support “*mwidth*” and “*weightm*”, e.g., `mec` to mean “go to medium weight and extra-condensed width”. At the end of the process we automatically drop any leftover `m` in the series name (unless it is just a single `m`).
- If there is no table entry then the target series is used unconditionally. This means that any request to set both weight and width (e.g. `bx` or `ulc`) needs no table entries. For that reason there are no entries which have a `weight+width` as request (i.e., second argument).

In particular this is also true for cases involving `m`, e.g., `bm` (bold medium width) which automatically gets reduced result in `b` or `mc` (medium weight condensed) which becomes `c` as a result.

- Only a few entries have “alterative” values and perhaps most of them should get dropped. Or maybe not . . . needs some thought perhaps.

The idea is that you don't want the normal substitution to kick in because that would reset the shape first and it may be better to stay with `b` when a change to `c` is requested and `bc` doesn't exist, than to go to first change the shape to `n` and then find that `bc/n` doesn't exist either and thus ending up with `m/n`.

- Also: while I did set up all nine standard weight values from `ul` to `ub` I only bothered to provide entries for `ec`, `sc`, `c` and `x`, because other levels of compression/expansion are not in any real fonts that I know.

Could and perhaps should be eventually extended to cover the whole set.

```
8 \DeclareFontSeriesChangeRule {bc}{b}{bc}{}
9 \DeclareFontSeriesChangeRule {bc}{c}{bc}{}
10 \DeclareFontSeriesChangeRule {bc}{eb}{ebc}{}
11 \DeclareFontSeriesChangeRule {bc}{ec}{bec} {bc}
12 \DeclareFontSeriesChangeRule {bc}{el}{elc}{}
13 \DeclareFontSeriesChangeRule {bc}{l}{lc}{}
14 \DeclareFontSeriesChangeRule {bc}{sb}{sbc}{}
15 \DeclareFontSeriesChangeRule {bc}{sc}{bsc} {bc}
16 \DeclareFontSeriesChangeRule {bc}{sl}{slc}{}
17 \DeclareFontSeriesChangeRule {bc}{ub}{ubc}{}
18 \DeclareFontSeriesChangeRule {bc}{ul}{ulc}{}
19 \DeclareFontSeriesChangeRule {bc}{x}{bx}{}

```

```

20 \DeclareFontSeriesChangeRule {bx}{b}{bx}{}
21 \DeclareFontSeriesChangeRule {bx}{c} {bc} {bx} %<-----
22 \DeclareFontSeriesChangeRule {bx}{eb}{ebx}{}
23 \DeclareFontSeriesChangeRule {bx}{ec} {bec} {bx} %<-----
24 \DeclareFontSeriesChangeRule {bx}{el}{elx}{}
25 \DeclareFontSeriesChangeRule {bx}{l}{lx}{}
26 \DeclareFontSeriesChangeRule {bx}{sb} {sbx} {}
27 \DeclareFontSeriesChangeRule {bx}{sc} {bsc} {bx} %<-----
28 \DeclareFontSeriesChangeRule {bx}{sl}{slx} {}
29 \DeclareFontSeriesChangeRule {bx}{ub}{ubx}{}
30 \DeclareFontSeriesChangeRule {bx}{ul}{ulx}{}
31 \DeclareFontSeriesChangeRule {bx}{x}{bx}{}

32 \DeclareFontSeriesChangeRule {b}{bx} {bx} {b} %<-----
33 \DeclareFontSeriesChangeRule {b}{c} {bc} {b} %<-----
34 \DeclareFontSeriesChangeRule {b}{ec} {bec} {b} %<-----
35 \DeclareFontSeriesChangeRule {b}{sb} {sb} {b} %<-----
36 \DeclareFontSeriesChangeRule {b}{sc} {bsc} {b} %<-----
37 \DeclareFontSeriesChangeRule {b}{x} {bx} {b} %<-----

38 \DeclareFontSeriesChangeRule {c}{bx} {bx} {b} %<-----
39 \DeclareFontSeriesChangeRule {c}{b}{bc}{}
40 \DeclareFontSeriesChangeRule {c}{eb}{ebc}{}
41 \DeclareFontSeriesChangeRule {c}{el}{elc}{}
42 \DeclareFontSeriesChangeRule {c}{l}{lc}{}
43 \DeclareFontSeriesChangeRule {c}{sb}{sbc}{}
44 \DeclareFontSeriesChangeRule {c}{sl}{slc}{}
45 \DeclareFontSeriesChangeRule {c}{ub}{ubc}{}
46 \DeclareFontSeriesChangeRule {c}{ul}{ulc}{}
47 \DeclareFontSeriesChangeRule {c}{x}{x}{m} %<-----

48 \DeclareFontSeriesChangeRule {ebc}{b}{bc}{}
49 \DeclareFontSeriesChangeRule {ebc}{c}{ebc}{}
50 \DeclareFontSeriesChangeRule {ebc}{eb}{ebc}{}
51 \DeclareFontSeriesChangeRule {ebc}{ec}{ebec}{ebc}
52 \DeclareFontSeriesChangeRule {ebc}{el}{elc}{}
53 \DeclareFontSeriesChangeRule {ebc}{l}{lc}{}
54 \DeclareFontSeriesChangeRule {ebc}{sb}{sbc}{}
55 \DeclareFontSeriesChangeRule {ebc}{sc}{ebsc}{ebc}
56 \DeclareFontSeriesChangeRule {ebc}{sl}{slc}{}
57 \DeclareFontSeriesChangeRule {ebc}{ub}{ubc}{}
58 \DeclareFontSeriesChangeRule {ebc}{ul}{ulc}{}
59 \DeclareFontSeriesChangeRule {ebc}{x}{ebx}{}

60 \DeclareFontSeriesChangeRule {ec}{bx} {bx} {b} %<-----
61 \DeclareFontSeriesChangeRule {ec}{b}{bec}{}
62 \DeclareFontSeriesChangeRule {ec}{eb}{ebec}{}
63 \DeclareFontSeriesChangeRule {ec}{el}{elec}{}
64 \DeclareFontSeriesChangeRule {ec}{l}{lec}{}
65 \DeclareFontSeriesChangeRule {ec}{sb}{sbec}{}
66 \DeclareFontSeriesChangeRule {ec}{sl}{slec}{}
67 \DeclareFontSeriesChangeRule {ec}{ub}{ubec}{}
68 \DeclareFontSeriesChangeRule {ec}{ul}{ulec}{}
69 \DeclareFontSeriesChangeRule {ec}{x}{x}{m} %<-----

70 \DeclareFontSeriesChangeRule {sc}{bx} {bx} {b} %<-----
71 \DeclareFontSeriesChangeRule {sc}{b}{bsc}{}

```

```

72 \DeclareFontSeriesChangeRule {sc}{eb}{ebsc}{}
73 \DeclareFontSeriesChangeRule {sc}{el}{elsc}{}
74 \DeclareFontSeriesChangeRule {sc}{l}{lsc}{}
75 \DeclareFontSeriesChangeRule {sc}{sb}{sbsc}{}
76 \DeclareFontSeriesChangeRule {sc}{sl}{slsc}{}
77 \DeclareFontSeriesChangeRule {sc}{ub}{ubsc}{}
78 \DeclareFontSeriesChangeRule {sc}{ul}{ulsc}{}
79 \DeclareFontSeriesChangeRule {sc}{x}{x}{m} %<-----

80 \DeclareFontSeriesChangeRule {ebx}{b}{bx}{}
81 \DeclareFontSeriesChangeRule {ebx}{c}{ebc}{}
82 \DeclareFontSeriesChangeRule {ebx}{eb}{ebx}{}
83 \DeclareFontSeriesChangeRule {ebx}{ec}{ebec}{}
84 \DeclareFontSeriesChangeRule {ebx}{el}{elx}{}
85 \DeclareFontSeriesChangeRule {ebx}{l}{lx}{}
86 \DeclareFontSeriesChangeRule {ebx}{sb}{sbx}{}
87 \DeclareFontSeriesChangeRule {ebx}{sc}{ebsc}{}
88 \DeclareFontSeriesChangeRule {ebx}{sl}{slx}{}
89 \DeclareFontSeriesChangeRule {ebx}{ub}{ubx}{}
90 \DeclareFontSeriesChangeRule {ebx}{ul}{ulx}{}
91 \DeclareFontSeriesChangeRule {ebx}{x}{ebx}{}

92 \DeclareFontSeriesChangeRule {eb}{c}{ebc}{}
93 \DeclareFontSeriesChangeRule {eb}{ec}{ebec}{}
94 \DeclareFontSeriesChangeRule {eb}{sc}{ebsc}{}
95 \DeclareFontSeriesChangeRule {eb}{x}{ebx}{}

96 \DeclareFontSeriesChangeRule {elc}{b}{bc}{}
97 \DeclareFontSeriesChangeRule {elc}{c}{elc}{}
98 \DeclareFontSeriesChangeRule {elc}{eb}{ebc}{}
99 \DeclareFontSeriesChangeRule {elc}{ec}{elec}{}
100 \DeclareFontSeriesChangeRule {elc}{el}{elc}{}
101 \DeclareFontSeriesChangeRule {elc}{l}{lc}{}
102 \DeclareFontSeriesChangeRule {elc}{sb}{sbc}{}
103 \DeclareFontSeriesChangeRule {elc}{sc}{elsc}{}
104 \DeclareFontSeriesChangeRule {elc}{sl}{slc}{}
105 \DeclareFontSeriesChangeRule {elc}{ub}{ubc}{}
106 \DeclareFontSeriesChangeRule {elc}{ul}{ulc}{}
107 \DeclareFontSeriesChangeRule {elc}{x}{elx}{}

108 \DeclareFontSeriesChangeRule {elx}{b}{bx}{}
109 \DeclareFontSeriesChangeRule {elx}{c}{elc}{}
110 \DeclareFontSeriesChangeRule {elx}{eb}{ebx}{}
111 \DeclareFontSeriesChangeRule {elx}{ec}{elec}{}
112 \DeclareFontSeriesChangeRule {elx}{el}{elx}{}
113 \DeclareFontSeriesChangeRule {elx}{l}{lx}{}
114 \DeclareFontSeriesChangeRule {elx}{sb}{sbx}{}
115 \DeclareFontSeriesChangeRule {elx}{sc}{elsc}{}
116 \DeclareFontSeriesChangeRule {elx}{sl}{slx}{}
117 \DeclareFontSeriesChangeRule {elx}{ub}{ubx}{}
118 \DeclareFontSeriesChangeRule {elx}{ul}{ulx}{}
119 \DeclareFontSeriesChangeRule {elx}{x}{elx}{}

120 \DeclareFontSeriesChangeRule {el}{c}{elc}{}
121 \DeclareFontSeriesChangeRule {el}{ec}{elec}{}
122 \DeclareFontSeriesChangeRule {el}{sc}{elsc}{}
123 \DeclareFontSeriesChangeRule {el}{x}{elx}{}

```

```

124 \DeclareFontSeriesChangeRule {lc}{b}{bc}{}
125 \DeclareFontSeriesChangeRule {lc}{c}{lc}{}
126 \DeclareFontSeriesChangeRule {lc}{eb}{ebc}{}
127 \DeclareFontSeriesChangeRule {lc}{ec}{lec}{}
128 \DeclareFontSeriesChangeRule {lc}{el}{elc}{}
129 \DeclareFontSeriesChangeRule {lc}{l}{lc}{}
130 \DeclareFontSeriesChangeRule {lc}{sb}{sbc}{}
131 \DeclareFontSeriesChangeRule {lc}{sc}{lsc}{}
132 \DeclareFontSeriesChangeRule {lc}{sl}{slc}{}
133 \DeclareFontSeriesChangeRule {lc}{ub}{ubc}{}
134 \DeclareFontSeriesChangeRule {lc}{ul}{ulc}{}
135 \DeclareFontSeriesChangeRule {lc}{x}{lx}{}

136 \DeclareFontSeriesChangeRule {lx}{b}{bx}{}
137 \DeclareFontSeriesChangeRule {lx}{c}{lc}{}
138 \DeclareFontSeriesChangeRule {lx}{eb}{ebx}{}
139 \DeclareFontSeriesChangeRule {lx}{ec}{lec}{}
140 \DeclareFontSeriesChangeRule {lx}{el}{elx}{}
141 \DeclareFontSeriesChangeRule {lx}{l}{lx}{}
142 \DeclareFontSeriesChangeRule {lx}{sb}{sbx}{}
143 \DeclareFontSeriesChangeRule {lx}{sc}{lsc}{}
144 \DeclareFontSeriesChangeRule {lx}{sl}{slx}{}
145 \DeclareFontSeriesChangeRule {lx}{ub}{ubx}{}
146 \DeclareFontSeriesChangeRule {lx}{ul}{ulx}{}
147 \DeclareFontSeriesChangeRule {lx}{x}{lx}{}

148 \DeclareFontSeriesChangeRule {l}{bx} {bx} {b} %<-----
149 \DeclareFontSeriesChangeRule {l}{b} {b} {bx} %<-----
150 \DeclareFontSeriesChangeRule {l}{c} {lc} {l} % ? %<-----
151 \DeclareFontSeriesChangeRule {l}{ec} {lec} {l} % ? %<-----
152 \DeclareFontSeriesChangeRule {l}{sb} {sb} {b} % ? %<-----
153 \DeclareFontSeriesChangeRule {l}{sc} {lsc} {l} % ? %<-----
154 \DeclareFontSeriesChangeRule {l}{x} {lx} {l} % ? %<-----

155 \DeclareFontSeriesChangeRule {m}{bx} {bx} {b} %<-----
156 \DeclareFontSeriesChangeRule {m}{b} {b} {bx} %<-----
157 \DeclareFontSeriesChangeRule {m}{c} {c} {m} %<-----
158 \DeclareFontSeriesChangeRule {m}{ec} {ec} {m} %<-----
159 \DeclareFontSeriesChangeRule {m}{l} {l} {m} %<-----
160 \DeclareFontSeriesChangeRule {m}{sb} {sb} {b} %<-----
161 \DeclareFontSeriesChangeRule {m}{sc} {sc} {m} %<-----
162 \DeclareFontSeriesChangeRule {m}{x} {x} {m} %<-----

163 \DeclareFontSeriesChangeRule {sbc}{b}{bc}{}
164 \DeclareFontSeriesChangeRule {sbc}{c}{sbc}{}
165 \DeclareFontSeriesChangeRule {sbc}{eb}{ebc}{}
166 \DeclareFontSeriesChangeRule {sbc}{ec}{sbec}{sbc}
167 \DeclareFontSeriesChangeRule {sbc}{el}{elc}{}
168 \DeclareFontSeriesChangeRule {sbc}{l}{lc}{}
169 \DeclareFontSeriesChangeRule {sbc}{sb}{sbc}{}
170 \DeclareFontSeriesChangeRule {sbc}{sc}{sbsc}{sbc}
171 \DeclareFontSeriesChangeRule {sbc}{sl}{slc}{}
172 \DeclareFontSeriesChangeRule {sbc}{ub}{ubc}{}
173 \DeclareFontSeriesChangeRule {sbc}{ul}{ulc}{}
174 \DeclareFontSeriesChangeRule {sbc}{x}{sbx}{}

175 \DeclareFontSeriesChangeRule {sbc}{b}{bc}{}

```

```

176 \DeclareFontSeriesChangeRule {sbx}{c}{sbc}{ }
177 \DeclareFontSeriesChangeRule {sbx}{eb}{ebx}{ }
178 \DeclareFontSeriesChangeRule {sbx}{ec}{sbec}{ }
179 \DeclareFontSeriesChangeRule {sbx}{el}{elx}{ }
180 \DeclareFontSeriesChangeRule {sbx}{l}{lx}{ }
181 \DeclareFontSeriesChangeRule {sbx}{sb}{sbs}{ }
182 \DeclareFontSeriesChangeRule {sbx}{sc}{sbsc}{ }
183 \DeclareFontSeriesChangeRule {sbx}{sl}{slx}{ }
184 \DeclareFontSeriesChangeRule {sbx}{ub}{ubx}{ }
185 \DeclareFontSeriesChangeRule {sbx}{ul}{ulx}{ }
186 \DeclareFontSeriesChangeRule {sbx}{x}{sbsx}{ }

187 \DeclareFontSeriesChangeRule {sb}{c} {sbc} {bc} %? %<-----
188 \DeclareFontSeriesChangeRule {sb}{ec} {sbec} {sbc} %? %<-----
189 \DeclareFontSeriesChangeRule {sb}{sc} {sbsc} {sbc} %? %<-----
190 \DeclareFontSeriesChangeRule {sb}{x} {sbsx} {bx} %? %<-----

191 \DeclareFontSeriesChangeRule {slc}{b}{bc}{ }
192 \DeclareFontSeriesChangeRule {slc}{c}{slc}{ }
193 \DeclareFontSeriesChangeRule {slc}{eb}{ebc}{ }
194 \DeclareFontSeriesChangeRule {slc}{ec}{slec}{ }
195 \DeclareFontSeriesChangeRule {slc}{el}{elc}{ }
196 \DeclareFontSeriesChangeRule {slc}{l}{lc}{ }
197 \DeclareFontSeriesChangeRule {slc}{sb}{sbc}{ }
198 \DeclareFontSeriesChangeRule {slc}{sc}{slsc}{ }
199 \DeclareFontSeriesChangeRule {slc}{sl}{slc}{ }
200 \DeclareFontSeriesChangeRule {slc}{ub}{ubc}{ }
201 \DeclareFontSeriesChangeRule {slc}{ul}{ulc}{ }
202 \DeclareFontSeriesChangeRule {slc}{x}{slx}{ }

203 \DeclareFontSeriesChangeRule {slx}{b}{bx}{ }
204 \DeclareFontSeriesChangeRule {slx}{c}{slc}{ }
205 \DeclareFontSeriesChangeRule {slx}{eb}{ebx}{ }
206 \DeclareFontSeriesChangeRule {slx}{ec}{slec}{ }
207 \DeclareFontSeriesChangeRule {slx}{el}{elx}{ }
208 \DeclareFontSeriesChangeRule {slx}{l}{lx}{ }
209 \DeclareFontSeriesChangeRule {slx}{sb}{sbs}{ }
210 \DeclareFontSeriesChangeRule {slx}{sc}{slsc}{ }
211 \DeclareFontSeriesChangeRule {slx}{sl}{slx}{ }
212 \DeclareFontSeriesChangeRule {slx}{ub}{ubx}{ }
213 \DeclareFontSeriesChangeRule {slx}{ul}{ulx}{ }
214 \DeclareFontSeriesChangeRule {slx}{x}{slx}{ }

215 \DeclareFontSeriesChangeRule {sl}{c}{slc}{ }
216 \DeclareFontSeriesChangeRule {sl}{ec}{slec}{ }
217 \DeclareFontSeriesChangeRule {sl}{sc}{slsc}{ }
218 \DeclareFontSeriesChangeRule {sl}{x}{slx}{ }

219 \DeclareFontSeriesChangeRule {ubc}{b}{bc}{ }
220 \DeclareFontSeriesChangeRule {ubc}{c}{ubc}{ }
221 \DeclareFontSeriesChangeRule {ubc}{eb}{ebc}{ }
222 \DeclareFontSeriesChangeRule {ubc}{ec}{ubec}{ }
223 \DeclareFontSeriesChangeRule {ubc}{el}{elc}{ }
224 \DeclareFontSeriesChangeRule {ubc}{l}{lc}{ }
225 \DeclareFontSeriesChangeRule {ubc}{sb}{sbc}{ }
226 \DeclareFontSeriesChangeRule {ubc}{sc}{ubsc}{ }
227 \DeclareFontSeriesChangeRule {ubc}{sl}{slc}{ }

```

228 \DeclareFontSeriesChangeRule {ubc}{ub}{ubc}{}
229 \DeclareFontSeriesChangeRule {ubc}{ul}{ulc}{}
230 \DeclareFontSeriesChangeRule {ubc}{x}{ubx}{}

231 \DeclareFontSeriesChangeRule {ubx}{b}{bx}{}
232 \DeclareFontSeriesChangeRule {ubx}{c}{uc}{}
233 \DeclareFontSeriesChangeRule {ubx}{eb}{ebx}{}
234 \DeclareFontSeriesChangeRule {ubx}{ec}{ubec}{}
235 \DeclareFontSeriesChangeRule {ubx}{el}{elx}{}
236 \DeclareFontSeriesChangeRule {ubx}{l}{lx}{}
237 \DeclareFontSeriesChangeRule {ubx}{sb}{sbx}{}
238 \DeclareFontSeriesChangeRule {ubx}{sc}{ubsc}{}
239 \DeclareFontSeriesChangeRule {ubx}{sl}{slx}{}
240 \DeclareFontSeriesChangeRule {ubx}{ub}{ubx}{}
241 \DeclareFontSeriesChangeRule {ubx}{ul}{ulx}{}
242 \DeclareFontSeriesChangeRule {ubx}{x}{ubx}{}

243 \DeclareFontSeriesChangeRule {ub}{c}{ubc}{}
244 \DeclareFontSeriesChangeRule {ub}{ec}{ubec}{}
245 \DeclareFontSeriesChangeRule {ub}{sc}{ubsc}{}
246 \DeclareFontSeriesChangeRule {ub}{x}{ubx}{}

247 \DeclareFontSeriesChangeRule {ulc}{b}{bc}{}
248 \DeclareFontSeriesChangeRule {ulc}{c}{ulc}{}
249 \DeclareFontSeriesChangeRule {ulc}{eb}{ebc}{}
250 \DeclareFontSeriesChangeRule {ulc}{ec}{ulec}{ulc}
251 \DeclareFontSeriesChangeRule {ulc}{el}{elc}{}
252 \DeclareFontSeriesChangeRule {ulc}{l}{lc}{}
253 \DeclareFontSeriesChangeRule {ulc}{sb}{sbc}{}
254 \DeclareFontSeriesChangeRule {ulc}{sc}{ulsc}{ulc}
255 \DeclareFontSeriesChangeRule {ulc}{sl}{slc}{}
256 \DeclareFontSeriesChangeRule {ulc}{ub}{ubc}{}
257 \DeclareFontSeriesChangeRule {ulc}{ul}{ulc}{}
258 \DeclareFontSeriesChangeRule {ulc}{x}{ulx}{}

259 \DeclareFontSeriesChangeRule {ulx}{b}{bx}{}
260 \DeclareFontSeriesChangeRule {ulx}{c}{ulc}{}
261 \DeclareFontSeriesChangeRule {ulx}{eb}{ebx}{}
262 \DeclareFontSeriesChangeRule {ulx}{ec}{ulec}{}
263 \DeclareFontSeriesChangeRule {ulx}{el}{elx}{}
264 \DeclareFontSeriesChangeRule {ulx}{l}{lx}{}
265 \DeclareFontSeriesChangeRule {ulx}{sb}{sbx}{}
266 \DeclareFontSeriesChangeRule {ulx}{sc}{ulsc}{}
267 \DeclareFontSeriesChangeRule {ulx}{sl}{slx}{}
268 \DeclareFontSeriesChangeRule {ulx}{ub}{ubx}{}
269 \DeclareFontSeriesChangeRule {ulx}{ul}{ulx}{}
270 \DeclareFontSeriesChangeRule {ulx}{x}{ulx}{}

271 \DeclareFontSeriesChangeRule {ul}{c}{ulc}{}
272 \DeclareFontSeriesChangeRule {ul}{ec}{ulec}{}
273 \DeclareFontSeriesChangeRule {ul}{sc}{ulsc}{}
274 \DeclareFontSeriesChangeRule {ul}{x}{ulx}{}

275 \DeclareFontSeriesChangeRule {x}{b}{bx}{}
276 \DeclareFontSeriesChangeRule {x}{c}{c}{}
277 \DeclareFontSeriesChangeRule {x}{eb}{ebx}{}
278 \DeclareFontSeriesChangeRule {x}{ec}{ec}{}
279 \DeclareFontSeriesChangeRule {x}{el}{elx}{}

```

280 \DeclareFontSeriesChangeRule {x}{l}{lx}{}
281 \DeclareFontSeriesChangeRule {x}{sb}{sxb}{}
282 \DeclareFontSeriesChangeRule {x}{sc}{sc}{}
283 \DeclareFontSeriesChangeRule {x}{sl}{slx}{}
284 \DeclareFontSeriesChangeRule {x}{ub}{ubx}{}
285 \DeclareFontSeriesChangeRule {x}{ul}{ulx}{}

```

Special rules for `lm` etc. aren't needed because if the target `lm` is request it will be used if there is no rule and that id then reduced to `l` automatically. Same for `mc` and friends. Only `?m` and `m?` need rules.

So here are the special rules for `m?`:

```

286 \DeclareFontSeriesChangeRule {bc}{m?}{c}{}
287 \DeclareFontSeriesChangeRule {bec}{m?}{ec}{}
288 \DeclareFontSeriesChangeRule {bsc}{m?}{sc}{}
289 \DeclareFontSeriesChangeRule {bx}{m?}{x}{}
290 \DeclareFontSeriesChangeRule {b}{m?}{m}{}
291 \DeclareFontSeriesChangeRule {c}{m?}{c}{}
292 \DeclareFontSeriesChangeRule {ebc}{m?}{c}{}
293 \DeclareFontSeriesChangeRule {ebec}{m?}{ec}{}
294 \DeclareFontSeriesChangeRule {ebsc}{m?}{sc}{}
295 \DeclareFontSeriesChangeRule {ebx}{m?}{x}{}
296 \DeclareFontSeriesChangeRule {eb}{m?}{m}{}
297 \DeclareFontSeriesChangeRule {ec}{m?}{ec}{}
298 \DeclareFontSeriesChangeRule {elc}{m?}{c}{}
299 \DeclareFontSeriesChangeRule {elec}{m?}{ec}{}
300 \DeclareFontSeriesChangeRule {elsc}{m?}{sc}{}
301 \DeclareFontSeriesChangeRule {elx}{m?}{x}{}
302 \DeclareFontSeriesChangeRule {el}{m?}{m}{}
303 \DeclareFontSeriesChangeRule {lc}{m?}{c}{}
304 \DeclareFontSeriesChangeRule {lec}{m?}{ec}{}
305 \DeclareFontSeriesChangeRule {lsc}{m?}{sc}{}
306 \DeclareFontSeriesChangeRule {lx}{m?}{x}{}
307 \DeclareFontSeriesChangeRule {l}{m?}{m}{}
308 \DeclareFontSeriesChangeRule {m}{m?}{m}{}
309 \DeclareFontSeriesChangeRule {sbc}{m?}{c}{}
310 \DeclareFontSeriesChangeRule {sbec}{m?}{ec}{}
311 \DeclareFontSeriesChangeRule {bsc}{m?}{sc}{}
312 \DeclareFontSeriesChangeRule {sxb}{m?}{x}{}
313 \DeclareFontSeriesChangeRule {sb}{m?}{m}{}
314 \DeclareFontSeriesChangeRule {sc}{m?}{sc}{}
315 \DeclareFontSeriesChangeRule {slc}{m?}{c}{}
316 \DeclareFontSeriesChangeRule {slec}{m?}{ec}{}
317 \DeclareFontSeriesChangeRule {slsc}{m?}{sc}{}
318 \DeclareFontSeriesChangeRule {slx}{m?}{x}{}
319 \DeclareFontSeriesChangeRule {sl}{m?}{m}{}
320 \DeclareFontSeriesChangeRule {ubc}{m?}{c}{}
321 \DeclareFontSeriesChangeRule {ubec}{m?}{ec}{}
322 \DeclareFontSeriesChangeRule {ubsc}{m?}{sc}{}
323 \DeclareFontSeriesChangeRule {ubx}{m?}{x}{}
324 \DeclareFontSeriesChangeRule {ub}{m?}{ub}{}
325 \DeclareFontSeriesChangeRule {ulc}{m?}{c}{}
326 \DeclareFontSeriesChangeRule {ulec}{m?}{ec}{}
327 \DeclareFontSeriesChangeRule {ulsc}{m?}{sc}{}
328 \DeclareFontSeriesChangeRule {ulx}{m?}{x}{}

```

```

329 \DeclareFontSeriesChangeRule {ul}{m?}{m}{}
330 \DeclareFontSeriesChangeRule {x}{m?}{x}{}
    And there the special rules for ?m:
331 \DeclareFontSeriesChangeRule {bc}{?m}{b}{}
332 \DeclareFontSeriesChangeRule {bec}{?m}{b}{}
333 \DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
334 \DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
335 \DeclareFontSeriesChangeRule {bx}{?m}{b}{}
336 \DeclareFontSeriesChangeRule {b}{?m}{b}{}
337 \DeclareFontSeriesChangeRule {c}{?m}{m}{}
338 \DeclareFontSeriesChangeRule {ebc}{?m}{eb}{}
339 \DeclareFontSeriesChangeRule {ebec}{?m}{eb}{}
340 \DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
341 \DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
342 \DeclareFontSeriesChangeRule {ebx}{?m}{eb}{}
343 \DeclareFontSeriesChangeRule {eb}{?m}{eb}{}
344 \DeclareFontSeriesChangeRule {ec}{?m}{m}{}
345 \DeclareFontSeriesChangeRule {elc}{?m}{el}{}
346 \DeclareFontSeriesChangeRule {elec}{?m}{el}{}
347 \DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
348 \DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
349 \DeclareFontSeriesChangeRule {elx}{?m}{el}{}
350 \DeclareFontSeriesChangeRule {el}{?m}{el}{}
351 \DeclareFontSeriesChangeRule {lc}{?m}{l}{}
352 \DeclareFontSeriesChangeRule {lec}{?m}{l}{}
353 \DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
354 \DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
355 \DeclareFontSeriesChangeRule {lx}{?m}{l}{}
356 \DeclareFontSeriesChangeRule {l}{?m}{l}{}
357 \DeclareFontSeriesChangeRule {m}{?m}{m}{}
358 \DeclareFontSeriesChangeRule {sbc}{?m}{sb}{}
359 \DeclareFontSeriesChangeRule {sbec}{?m}{sb}{}
360 \DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
361 \DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
362 \DeclareFontSeriesChangeRule {sbx}{?m}{sb}{}
363 \DeclareFontSeriesChangeRule {sb}{?m}{sb}{}
364 \DeclareFontSeriesChangeRule {sc}{?m}{m}{}
365 \DeclareFontSeriesChangeRule {sc}{?m}{m}{}
366 \DeclareFontSeriesChangeRule {slc}{?m}{sl}{}
367 \DeclareFontSeriesChangeRule {slec}{?m}{sl}{}
368 \DeclareFontSeriesChangeRule {slsc}{?m}{sl}{}
369 \DeclareFontSeriesChangeRule {slsc}{?m}{sl}{}
370 \DeclareFontSeriesChangeRule {slx}{?m}{sl}{}
371 \DeclareFontSeriesChangeRule {sl}{?m}{sl}{}
372 \DeclareFontSeriesChangeRule {ubc}{?m}{ub}{}
373 \DeclareFontSeriesChangeRule {ubec}{?m}{ub}{}
374 \DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
375 \DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
376 \DeclareFontSeriesChangeRule {ubx}{?m}{ub}{}
377 \DeclareFontSeriesChangeRule {ub}{?m}{m}{}
378 \DeclareFontSeriesChangeRule {ulc}{?m}{ul}{}
379 \DeclareFontSeriesChangeRule {ulec}{?m}{ul}{}
380 \DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}
381 \DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}

```

```

382 \DeclareFontSeriesChangeRule {ulx}{?m}{ul}{-}
383 \DeclareFontSeriesChangeRule {ul}{?m}{ul}{-}
384 \DeclareFontSeriesChangeRule {x}{?m}{m}{-}

```

28.3 Changing to a new series

`\if@forced@series` If the series gets forced we need to know that fact later on.

```
385 \newif\if@forced@series
```

`\fontseriesforce` To change unconditionally to a new series you can use `\fontseriesforce`. If course, if the series doesn't exist for the current family substitution still happens, but there is not dependency on the current series.

```
386 \DeclareRobustCommand\fontseriesforce[1]{\@forced@seriestrue\edef\f@series{#1}}
```

`\fontseries` The `\fontseries` command takes one argument which is the requested new font series. In the original implementation it simply saved the expanded value in `\f@series`. Now we do a bit more processing and look up the final value in the font series data base. This is done by `\merge@font@series`.

```
387 \DeclareRobustCommand\fontseries[1]{\@forced@seriesfalse\merge@font@series{#1}}
```

`\merge@font@series` We look up the data base value by expanding the right command twice. If no such value exist then the result will be `\relax` otherwise it will be the two brace groups: the desired result and the alternate result. The first case means that the third argument to `\merge@font@series` will be empty.

```

388 \def\merge@font@series#1{%
389   \expandafter\expandafter\expandafter
390   \merge@font@series@
391   \csname series@\f@series @#1\endcsname
392   {#1}%
393   \@nil
394 }

```

`\merge@font@series@` This now defines the new `\f@series`:

```
395 \def\merge@font@series@#1#2#3\@nil{%
```

If the third argument is empty there is no database entry for the combination and the second argument holds the new series so we return that.

Originally the test was simply `\ifx!#3!` but that actually dies if `#3` starts with a conditional and in the definition of `\AmSfont` that is actually the case.

```

396 %\ifcat\expandafter X\detokenize{#1}X%
397 \def\reserved@a{#3}%
398 \ifx\reserved@a\@empty
399   \set@target@series{#2}%
400 \else

```

Otherwise we check if the desired result for the series (`#1`) exists for the font family and the current shape. As the `.fd` is perhaps not loaded yet, we first have to do that, otherwise the test would fail even if the face is actually available.

```

401   \maybe@load@fontshape
402   \edef\reserved@a{\f@encoding /\f@family /#1/\f@shape}%
403   \ifcsname \reserved@a \endcsname

```

If the desired result is available then we use that. However, we do need some post-processing because we need to drop surplus ms due to the way naming convention was designed in the '90s (sigh).

```
404     \set@target@series{#1}%
```

If not, then we try the alternate result (#2).

```
405     \else
```

```
406     \ifcsname \f@encoding /\f@family /#2/\f@shape \endcsname
```

If the alternate result exist we use that and also issue a warning (or rather a log entry) that we didn't managed to change to the desired font.

```
407     \set@target@series{#2}%
```

```
408     \@font@shape@subst@warning
```

If that doesn't exist either, then we use the requested series unmodified (again with a warning).

```
409     \else
```

```
410     \set@target@series{#3}%
```

```
411     \@font@shape@subst@warning
```

```
412     \fi
```

```
413     \fi
```

```
414 \fi
```

```
415 }
```

It is possible that the previous font and the new one are actually identical (and the font was not found because it still needs loading) in which case a warning would look rather odd. So we make a quick check for that (which is the reason why we defined \@reserveda above.

```
416 \def\@font@shape@subst@warning{%
```

```
417   \edef\reserved@b{\curr@fontshape}%
```

```
418   \ifx\reserved@a\reserved@b \else
```

```
419     \@font@warning{Font shape '\reserved@a' undefined\MessageBreak
420                   using '\reserved@b' instead}%
```

```
421   \fi
```

```
422 }
```

`\maybe@load@fontshape` A small helper that we use a couple of times: try loading a fontshape (in a group because `\try@load@fontshape` normalizes catcodes).

```
423 \def\maybe@load@fontshape{\begingroup\try@load@fontshape\endgroup}
```

`\set@target@series` Finally the code for normalizing the `\f@series` value.

The combined series value determined by the mapping may still contain an m that we have to remove (as the .fd files use c not mc to denote a medium weight condensed series, etc.). We do this in all branches above because a user might have written

```
\DeclareFontSeriesChangeRule {m}{sc}{msc}{mc}
```

instead of using sc and c as needed in the .fd file.

```
424 \def\set@target@series#1{%
```

We need to \edef the argument first in case it starts with a conditional. Then we check (and perhaps drop) an "m" from the value and assign the result to `\f@series`.

```
425   \edef\f@series{#1}%
```

```
426 \expandafter\series@maybe@drop@one@m\expandafter{\f@series}\f@series
427 }
```

`\series@maybe@drop@one@m` If the series value is in NFSS notation then it should not contain any “m” unless it is just an “m” by it own. So we need to drop surplus “m”s. But we better don’t do this for full names, such as “semibold” as used by `autoinst`, for example. So we test against the possible explicit values that should drop an “m”. After that we assign the result to #2 for further use.

```
428 \def\series@maybe@drop@one@m#1#2{%
```

The code below is an inline version of the `\in@` macro without the group, so that it works in `\accent`.

```
429 \def\in@##1,#1,}%
430 \series@check@toks\expandafter{\in@@
431 ,ulm,elm,lm,slm,mm,sbm,bm,ebm,ubm,muc,mec,mc,msc,msx,mx,mex,mux,{}{},#1,}%
432 \edef\in@{\the\series@check@toks}%
433 \ifx\in@\@empty
434 \def#2{#1}%
435 \else
436 \edef#2{\expandafter\series@drop@one@m #1m\series@drop@one@m}%
437 \fi
438 }
```

As a precaution we use a private toks register not `\toks@` as that is no longer hidden inside the group.

```
439 \newtoks\series@check@toks
```

`\series@drop@one@m` Drop up to two ms but keep one if that makes the series value empty. Actually, with the current implementation we know that there is at least one in the series value itself and we added one after it, so all we have to do is now returning #1#2 and dropping the rest.

```
440 \def\series@drop@one@m#1#2#3\series@drop@one@m{%
441 % \ifx\relax#1#2\relax m\else#1#2\fi
442 #1#2%
443 }
```

29 Changing the shape

Shapes are also split in two axes (though it could be more if that is desirable), essentially building in an “sc” axis).

`\DeclareFontShapeChangeRule` The database for shapes is done in exactly the same way, only that it is much smaller and we usually have no alternative shape (or rather it is empty thus not used).

```
444 \def\DeclareFontShapeChangeRule #1#2#3#4{%
445 \@namedef{shape@#1@#2}{#{3}-#{4}}}
```

There is kind of the same problem with returning back from `sc` to normal. It sort of needs its own letter. In `fontspec` this was solved by the first time `\upshape` changes it or `sl` back (so only `sc` remains) and second time it changes then `sc` back to normal. Maybe that’s not a bad way to handle it, but decided for

a slightly different approach: `n` always returns to “normal”, ie resets everything and `up` changes italic or slanted to upright and `ulc` undoes small caps.

So we now offer `\normalshape` (using `\shapedefault` which is normally the same as calling both `\ulcshape` and `\upshape`, only more efficient.

`\ulcshape` To request going back to upper/lowercase we need a new command. It uses `ulc`
`\textulc` as shape name but this shape is virtual, i.e., it doesn't exist as a real shape, it
`\ulcdefault` is only used as part of the database table entries and thus only appears in the second argument there (but not in the first).

```
446 \DeclareRobustCommand\ulcshape
447     {\not@math@alphabet\ulcshape\relax
448     \fontshape\ulcdefault\selectfont}
449 \newcommand\ulcdefault{ulc}
```

`\swshape` New command to select a swash shape. The standard rules put this in the same
`\textsw` category as italics or slanted, i.e., if you ask for it then italics are undone. One
`\swdefault` could provide more complicated rules so that `it + sw` becomes `swit` but given that there are only very few fonts that have swash letters that level of flexibility (these days) would be just resulting in a lot of combinations that do not exist.

```
450 \DeclareRobustCommand\swshape
451     {\not@math@alphabet\swshape\relax
452     \fontshape\swdefault\selectfont}
453 \newcommand\swdefault{sw}
```

`\sscshape` New command to select spaced small capitals. This is only here because `fontaxes`
`\textssc` offered it. There isn't a single free font that supports it. However, some commercial
`\sscdefault` ones do, so we offer it so that at some point `fontaxes` could be retired.

So far there aren't any rules for it—probably there should be some putting it in the same category as `sc`.

```
454 \DeclareRobustCommand\sscshape
455     {\not@math@alphabet\sscshape\relax
456     \fontshape\sscdefault\selectfont}
457 \newcommand\sscdefault{ssc}
```

29.1 Mapping rules for shape combinations

Many of the entries are commented out as we will get that result without any entry.

```
458 %\DeclareFontShapeChangeRule {n}{n} {n} {}
459 \DeclareFontShapeChangeRule {n}{it} {it} {sl}
460 \DeclareFontShapeChangeRule {n}{sl} {sl} {it}
461 %\DeclareFontShapeChangeRule {n}{sw} {sw} {}
462 %\DeclareFontShapeChangeRule {n}{sc} {sc} {}
463 \DeclareFontShapeChangeRule {n}{ulc} {n} {}
464 \DeclareFontShapeChangeRule {n}{up} {n} {}

465 %\DeclareFontShapeChangeRule {it}{n} {n} {}
466 %\DeclareFontShapeChangeRule {it}{it} {it} {}
467 \DeclareFontShapeChangeRule {it}{sl} {sl} {it}
468 %\DeclareFontShapeChangeRule {it}{sw} {sw} {}
```

If neither `scit` nor `scsl` exist then `sc` will be used as a fallback albeit with a log entry, so except for the latter there will be no change for CM or Latin Modern fonts.

```

469 \DeclareFontShapeChangeRule {it}{sc} {scit} {scsl}
470 \DeclareFontShapeChangeRule {it}{ulc} {it} {}
471 \DeclareFontShapeChangeRule {it}{up} {n} {}

472 %\DeclareFontShapeChangeRule {sl}{n} {n} {}
473 \DeclareFontShapeChangeRule {sl}{it} {it} {sl}
474 %\DeclareFontShapeChangeRule {sl}{sl} {sl} {}
475 %\DeclareFontShapeChangeRule {sl}{sw} {sw} {}
476 \DeclareFontShapeChangeRule {sl}{sc} {scsl} {scit}
477 \DeclareFontShapeChangeRule {sl}{ulc} {sl} {}
478 \DeclareFontShapeChangeRule {sl}{up} {n} {}

479 %\DeclareFontShapeChangeRule {sc}{n} {n} {}
480 \DeclareFontShapeChangeRule {sc}{it} {scit} {scsl}
481 \DeclareFontShapeChangeRule {sc}{sl} {scsl} {scit}
482 \DeclareFontShapeChangeRule {sc}{sw} {scsw} {sw}
483 %\DeclareFontShapeChangeRule {sc}{sc} {sc} {}
484 \DeclareFontShapeChangeRule {sc}{ulc} {n} {}

```

The next rule might be a bit surprising and rightly so. Correct would be that `sc` is not affected by `up`, i.e., remains `sc` as showed in the commented out rule. However, for nearly three decades commands such as `sc` or `\textup` changed small caps back to the “normal” shape. So for backward compatibility we keep that behavior.

As a result you are currently typesetting in `scit` or `scsl` using `\upshape` twice will return you to the normal shape too, the first will change to `sc` and the second (because of the rule below) change that to `n`. This is the way `fontspec` implemented its version on this interface, so this rule means we are also compatible with the way `fontspec` behaved. Still it remains an oddity which I would rather liked to have avoided.

```

485 %\DeclareFontShapeChangeRule {sc}{up} {sc} {}
486 \DeclareFontShapeChangeRule {sc}{up} {n} {}

487 %\DeclareFontShapeChangeRule {scit}{n} {n} {}
488 \DeclareFontShapeChangeRule {scit}{it} {scit} {}
489 \DeclareFontShapeChangeRule {scit}{sl} {scsl} {scit}
490 \DeclareFontShapeChangeRule {scit}{sw} {scsw} {sc} % or scit?
491 \DeclareFontShapeChangeRule {scit}{sc} {scit} {}
492 \DeclareFontShapeChangeRule {scit}{ulc} {it} {}
493 \DeclareFontShapeChangeRule {scit}{up} {sc} {}

```

The previous rule assumes that if `scit` exists then `it` exists as well. If not, the mechanism will save `ulc` in `\f@series` which most certainly doesn’t exist. So when a font is later selected that would result in a substitution (so no harm done really). Alternatively, we could in this case use `n` as alternative, which may be a bit faster, but such a setup would be so weird in the first place that this isn’t worth the effort.

```

494 %\DeclareFontShapeChangeRule {scsl}{n} {n} {}
495 \DeclareFontShapeChangeRule {scsl}{it} {scit} {scsl}
496 \DeclareFontShapeChangeRule {scsl}{sl} {scsl} {}
497 \DeclareFontShapeChangeRule {scsl}{sw} {scsw} {sc} % or scsl?

```

```

498 \DeclareFontShapeChangeRule {scsl}{sc} {scsl} {}
499 \DeclareFontShapeChangeRule {scsl}{ulc} {sl} {}
500 \DeclareFontShapeChangeRule {scsl}{up} {sc} {}

501 %\DeclareFontShapeChangeRule {scsw}{n} {n} {}
502 \DeclareFontShapeChangeRule {scsw}{it} {scit} {scsw}
503 \DeclareFontShapeChangeRule {scsw}{sl} {scsl} {}
504 \DeclareFontShapeChangeRule {scsw}{sw} {scsw} {}
505 \DeclareFontShapeChangeRule {scsw}{sc} {scsw} {}
506 \DeclareFontShapeChangeRule {scsw}{ulc} {sw} {}
507 \DeclareFontShapeChangeRule {scsw}{up} {sc} {}

508 %\DeclareFontShapeChangeRule {sw}{n} {n} {}
509 %\DeclareFontShapeChangeRule {sw}{it} {it} {}
510 %\DeclareFontShapeChangeRule {sw}{sl} {sl} {}
511 %\DeclareFontShapeChangeRule {sw}{sw} {sw} {}
512 \DeclareFontShapeChangeRule {sw}{sc} {scsw} {}
513 \DeclareFontShapeChangeRule {sw}{ulc} {sw} {}
514 \DeclareFontShapeChangeRule {sw}{up} {n} {}

```

29.2 Changing to a new shape

`\fontshape` Again the `\fontshape` now has to do a lookup to get to its new value in `\f@shape`. The method is exactly the same as in `\fontseries`.

```
515 \DeclareRobustCommand\fontshape[1]{\merge@font@shape{#1}}
```

`\fontshapeforce` The unconditional version:

```
516 \DeclareRobustCommand\fontshapeforce[1]{\edef\f@shape{#1}}
```

`\merge@font@shape` Look up the database entry (if existing) and act accordingly.

```

517 \def\merge@font@shape#1{%
518   \expandafter\expandafter\expandafter
519   \merge@font@shape@
520   \csname shape@\f@shape @#1\endcsname
521   {#1}%
522   \@nil
523 }

```

`\merge@font@shape@` Same game now, except that we look at shapes not series values and we can set the shape without the complication of dropping “m”s from the name as we had to for the series.

```

524 \def\merge@font@shape@#1#2#3\@nil{%
525   \def\reserved@a{#3}%
526   \ifx\reserved@a\@empty
527     \edef\f@shape{#2}%
528   \else
529     \maybe@load@fontshape
530     \edef\reserved@a{\f@encoding /\f@family /\f@series/#1}%
531     \ifcsname \reserved@a\endcsname
532       \edef\f@shape{#1}%
533     \else
534       \ifcsname \f@encoding /\f@family /\f@series/#2\endcsname
535         \edef\f@shape{#2}%
536       \@font@shape@subst@warning

```

```

537     \else
538     \edef\f@shape{#3}%
539     \@font@shape@subst@warning
540     \fi
541     \fi
542     \fi
543 }

```

`\normalshape` `\normalshape` resets both sub-axes if the default rules are used.

```

544 \protected\def\normalshape
545   {\not@math@alphabet\normalshape\relax
546    \fontshape\shapedefault\selectfont}%

```

30 Make sure we win ...

This code implements one aspect of what the package `fontaxes` provide. So its redefinitions for the various shape commands, such as `\itshape` should no longer happen. We therefore force the standard definitions at `\AtBeginDocument` (later when this is defined. Once `fontaxes` is no longer doing such redefinitions that could be taken out again.

We use a separate macro so that we can easily disable this (in case of rollback).

`\reinstall@nfss@defs` I use `\protected` here not `\DeclareRobustCommand` to avoid extra status lines.

```

547 \def\reinstall@nfss@defs{%
548   \protected\def\upshape
549     {\not@math@alphabet\upshape\relax
550      \fontshape\updefault\selectfont}%
551   \protected\def\slshape
552     {\not@math@alphabet\slshape\relax
553      \fontshape\sldefault\selectfont}%
554   \protected\def\scshape
555     {\not@math@alphabet\scshape\relax
556      \fontshape\scdefault\selectfont}%
557   \protected\def\itshape
558     {\not@math@alphabet\itshape\mathit
559      \fontshape\itdefault\selectfont}%
560   \protected\def\ulcshape
561     {\not@math@alphabet\ulcshape\relax
562      \fontshape{ulc}\selectfont}%
563   \protected\def\swshape
564     {\not@math@alphabet\swshape\relax
565      \fontshape\swdefault\selectfont}%
566   \protected\def\sscshape
567     {\not@math@alphabet\sscshape\relax
568      \fontshape\sscdefault\selectfont}%
569 }

```

Supporting rollback ...

```

570 </2ekernel | latexrelease>
571 <latexrelease>\EndIncludeInRelease
572 <latexrelease>\IncludeInRelease{0000/00/00}%
573 <latexrelease>  {\DeclareFontSeriesChangeRule}{Series change rules}%

```

```

574 <latexrelease>
575 <latexrelease>\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
576 <latexrelease>\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
577 <latexrelease>\let\fontseriesforce\@undefined
578 <latexrelease>\let\fontshapeforce\@undefined
579 <latexrelease>
580 <latexrelease>\let\DeclareFontSeriesChangeRule\@undefined
581 <latexrelease>\let\merge@font@series\@undefined
582 <latexrelease>\let\merge@font@series@\@undefined
583 <latexrelease>\let\@font@shape@subst@warning\@undefined
584 <latexrelease>\let\maybe@load@fontshape\@undefined
585 <latexrelease>\let\set@target@series\@undefined
586 <latexrelease>\let\series@maybe@drop@one@m\@undefined
587 <latexrelease>\let\series@drop@one@m\@undefined
588 <latexrelease>\let\DeclareFontShapeChangeRule\@undefined
589 <latexrelease>\let\merge@font@shape\@undefined
590 <latexrelease>\let\merge@font@shape@\@undefined
591 <latexrelease>\let\normalshape\@undefined
592 <latexrelease>\let\ulcshape\@undefined
593 <latexrelease>\let\ulcdefault\@undefined
594 <latexrelease>\let\swshape\@undefined
595 <latexrelease>\let\swdefault\@undefined
596 <latexrelease>\let\sscshape\@undefined
597 <latexrelease>\let\sscdefault\@undefined
598 <latexrelease>\let\normalshape\@undefined

This is always called in \document so don't make it undefined.

599 <latexrelease>
600 <latexrelease>\let\reinstall@nfss@defs\relax
601 <latexrelease>\EndIncludeInRelease
602 <*2kernel>
603 </2kernel>

```

File q

lfsstrc.dtx

31 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 `tracefmt` 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 `tracefmt` 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 `tracefmt` package became active.

pausing Turn all font warnings into errors so that L^AT_EX will stop.

32 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{lfsstrc.dtx}
9 \end{document}
10 </driver>
```

33 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 \</package>
```

The debug module makes use of commands contained in a special package file named `trace.sty`.⁴

```
16 (+debug) \input trace.sty
```

34 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 (*2kernel)
18 \def\tracingfonts{%
19   \@font@warning{Command \noexpand\tracingfonts
20     not provided.\MessageBreak
21     Use the ‘tracefnt’ package.\MessageBreak Command found:}%
22   \count@}
23 \</2kernel>
```

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 \</package, trace, debug>
```

The option `errorshow` turns off all warnings so that only real errors are shown. `warningshow` corresponds to the NFSS default (when `tracefnt` is not loaded). `info` is the default for this package here; and `debugshow`, `loading`, and `pausing` extend the amount of information even further.

```
28 (*package)
29 \DeclareOption{errorshow}{%
30   \def\@font@info#1{%
31     \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
```

⁴This package is not in distribution at the moment (and probably doesn't any longer work). Think of this part of the code as being historical artefacts.

```

32             {LaTeX Font Info: \space\space\space#1}}%
33 \def\@font@warning#1{%
34     \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
35             {LaTeX Font Warning: #1}}%
36 }
37 \DeclareOption{warningshow}{%
38     \def\@font@info#1{%
39         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
40             {LaTeX Font Info: \space\space\space#1}}%
41     \def\@font@warning#1{%
42         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
43             {LaTeX Font Warning: #1}}%
44     }
45 \DeclareOption{infoshow}{%
46     \def\@font@info#1{%
47         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
48             {LaTeX Font Info: \space\space\space#1}}%
49     \def\@font@warning#1{%
50         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
51             {LaTeX Font Warning: #1}}%
52     }
53 \DeclareOption{loading}{%
54     \tracingfonts\tw@
55 }
56 \DeclareOption{debugshow}{%
57     \ExecuteOptions{infoshow}%
58     \tracingfonts\thr@@
59 }

60 \DeclareOption{pausing}{%
61     \def\@font@warning#1{%
62         \GenericError
63             {(Font)\@spaces\@spaces\@spaces\space\space}%
64             {LaTeX Font Warning: #1}%
65             {See the LaTeX Companion for details.}%
66             {I'll stop for every LaTeX Font Warning because
67             you requested\MessageBreak the 'pausing' option
68             to the tracefmt package.}}%
69     }

```

We make `infoshow` the default, which in turn defines `\font@warning` and `\font@info`.

```

70 \ExecuteOptions{infoshow}
71 \ProcessOptions
72 </package>

```

We also need a default definition inside the kernel:

```

73 <{*2kernel}
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)
```

35 Macros common to fam.tex and tracefnt.sty

In the first versions of `tracefnt.dtx` some macros of `fam.dtx`⁵ were redefined to include 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 `lftfss.dtx`.

35.1 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{%
```

⁵This file is currently not distributed in documented form. Its code is part of `lftfss.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 </trace>

```

Finally we call `\size@update`. This macro is normally empty but will contain actions (like setting the `\baselineskip`) that have to be carried out when the font size, the base `\baselineskip` or the `\baselinestretch` have changed.

```
128     \size@update
```

A similar function is called to handle anything related to encoding updates. This one is changed from `\relax` by `\fontencoding`.

```
129     \enc@update
```

Just before ending this macro we have to pop the tracing stack if it was pushed before.

```

130 <+debug> \poptracing
131     }

```

`\set@fontsize` The macro `\set@fontsize` does the actual work. First it assigns new values to `\f@size`, `\f@baselineskip` and `\f@linespread`.

```

132 \def\set@fontsize#1#2#3{%
133     \@defaultunits\@tempdimb#2pt\relax\@nnil
134     \edef\f@size{\strip@pt\@tempdimb}%
135     \@defaultunits\@tempskipa#3pt\relax\@nnil
136     \edef\f@baselineskip{\the\@tempskipa}%
137     \edef\f@linespread{#1}%

```

For backward compatibility and for later testing within `\selectfont` the internal value of `\f@linespread` is passed back to `\baselinestretch`.

```
138     \let\baselinestretch\f@linespread
```

Additional processing will happen within `\selectfont`. For this reason the macro `\size@update` (which will be called in `\selectfont`) will be defined to be:

```
139     \def\size@update{%
```

First calculate the new `\baselineskip` and also store it in `normalbaselineskip`

```

140         \baselineskip\f@baselineskip\relax
141         \baselineskip\f@linespread\baselineskip
142         \normalbaselineskip\baselineskip

```

then to set up a new `\strutbox`

```

143         \setbox\strutbox\hbox{%
144             \vrule\@height.7\baselineskip
145                 \@depth.3\baselineskip
146                 \@width\z@}%

```

We end with a bit of tracing information.

```

147 <*trace>
148     \ifnum \tracingfonts>\tw@
149         \ifx\f@linespread\@empty
150             \let\reserved@a\@empty
151         \else
152             \def\reserved@a{\f@linespread x}%
153         \fi
154         \font@info{Changing size to \f@size/\reserved@a
155                 \f@baselineskip}%
156         \aftergroup\type@restoreinfo \fi
157 </trace>

```

When all this is processed `\size@update` redefines itself to `\relax` so that in later calls of `\selectfont` no extra code will be executed.

```
158     \let\size@update\relax}%
159 }
```

Instead of defining this macro internally we might speed things up by placing the code into a separate macro and use `\let!`

`\size@update` Normally this macro does nothing; it will be redefined by `\set@fontsize` to initiate an update.

```
160 \let\size@update\relax
```

`\type@restoreinfo` This macro produces some info when a font size and/or baseline change will get restored.

```
161 (*trace)
162   \def\type@restoreinfo{%
163     \ifx\f@linespread\@empty
164       \let\reserved@a\@empty
165     \else
166       \def\reserved@a{\f@linespread x}%
167     \fi
168     \@font@info{Restoring size to
169                \f@size/\reserved@a\f@baselineskip}}
170 </trace>
```

`\glb@settings` The macro `\glb@settings` globally selects all math fonts for the current size if necessary.

```
171 \def\glb@settings{%
```

When `\glb@settings` gains control a size change was requested and all previous font assignments need to be replaced. Therefore the old values of the fonts are no longer needed. For every *math group* the new assignments are appended to `\math@fonts`. But this happens only if the `math@fonts` switch is set to true. However, we always set up the correct math sizes for script and scriptscript fonts since they may be needed even if we don't set up the whole math machinery.

Here we set the math size, script size and scriptscript size. If the `S@...` macro is not defined we have to first calculate the three sizes.

```
172   \expandafter\ifx\csname S@\f@size\endcsname\relax
173     \calculate@math@sizes
174   \fi
```

The effect of this is that `\calculate@math@sizes` may or may not define the `S@...` macro. In the first case the next time the same size is requested this macro is used, otherwise `\calculate@math@sizes` is called again. This also sets the `math@fonts` switch. If it is true we must switch the math fonts.

```
175   \csname S@\f@size\endcsname
176   \ifmath@fonts
177 (*trace)
178     \ifnum \tracingfonts>\tw@
179       \@font@info{Setting up math fonts for
180                  \f@size/\f@baselineskip}\fi
181 </trace>
```

Inside a group we execute the macro for the current math *version*. This sets `\math@fonts` to a list of `\textfont...` assignments. `\getanddefine@fonts` (which may be called at this point) needs the `\escapechar` parameter to be set to `-1`.

```
182     \begingroup
183     \escapechar\m@ne
184     \csname mv@\math@version \endcsname
```

Then we set `\globaldefs` to 1 so that all following changes are done globally. The math font assignments recorded in `\math@fonts` are executed and `\glb@currsiz` is set equal to `\f@size`. This signals that the fonts for math in this size are set up.

```
185     \globaldefs\@ne
186     \math@fonts
187     \let \glb@currsiz \f@size
188     \endgroup
```

Finally we execute any code that is supposed to happen whenever the math font setup changes. This register will be executed in local mode which means that everything that is supposed to have any effect should be done globally inside. We can't execute it within `\globaldefs\@ne` as we don't know what ends up inside this register, e.g., it might contain calculations which use some local registers to calculate the final (global) value.

```
189     \the\every@math@size
```

Otherwise we announce that the math fonts are not set up for this size.

```
190 (*trace)
191     \else
192     \ifnum \tracingfonts>\tw@
193     \font@info{No math setup for
194             \f@size/\f@baselineskip}\fi
195 </trace>
196     \fi
197 }
198 </2ekernel | 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 </2ekernel>
```

35.2 Math fonts setup

35.2.1 Outline of algorithm for math font sizes

\TeX uses the math fonts that are current when the end of a formula is reached. If we don't want to keep font setups local to every formula (which would result in

an enormous overhead, we have to be careful not to end up with the wrong setup in case formulas are nested, e.g., we need to be able to handle

```
$ a=b+c \mbox{ \small for all $b$ and $c\in Z$}$
```

Here the inner formulae `b` and `c\in Z` are typeset in `\small` but we have to return to `\normalsize` before we reach the closing `$` of the outer formula.

This is handled in the following way:

1. At any point in the document the global variable `\gbl@currsiz` contains the point size for which the math fonts currently are set up.
2. Whenever we start a formula we compare its value with the local variable `\f@size` that describes the current text font size.
3. If both are the same we assume that we can use the current math font setup without adjustment.
4. If they differ we call `\gbl@settings` which changes the math font setup and updates `\gbl@currsiz`.
 - (a) If we are recursively inside another formula (`\if@inmath`) we ensure that `\gbl@settings` is executed again in the outer formula, so that the old setup is automatically restored.
 - (b) Otherwise, we set the switch `@inmath` locally to `true` so that all nested formulae will be able to detect that they are nested in some outer formula.

The above algorithm has the following features:

- For sizes which are not containing any formula no math setup is done. Compared to the original algorithm of NFSS this results in the following savings:
 - No unnecessary loading of math fonts for sizes that are not used to typeset any math formulae (explicit or implicit ones).
 - No time overhead due to unnecessary changes of the math font setup on entrance and exit of the text font size.
- Math font setup changes for top-level formulae will survive (there is no restoration after the formula) thus any following formula in the same size will be directly typesettable. Compared to original implementation in NFSS2 the new algorithm has the overhead of one test per formula to see if the current math setup is valid (in the original algorithm the setup was always valid, thus no test was necessary).
- In nested formulae the math font setup is restored in the outer formula by a series of `\aftergroup` commands and checks. Compared to the original algorithm this involves additional checks ($2 \times \langle \text{non-math levels} \rangle$ per inner formula).

35.2.2 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@currsiz \f@size
207   (*trace)
208     \ifnum \tracingfonts>\thr@@
209       \@font@info{*** MATH: no change \f@size\space
210         curr/global (\curr@math@size/\glb@currsiz)}\fi
211   (/trace)
212   \else
213   (*trace)
214     \ifnum \tracingfonts>\thr@@
215       \@font@info{*** MATH: setting up \f@size\space
216         curr/global (\curr@math@size/\glb@currsiz)}\fi
217   (/trace)
218     \glb@settings
219     \init@restore@glb@settings
220   \fi
221   \let\curr@math@size\f@size
222   \def\init@restore@glb@settings{\aftergroup\restglb@settings}%
223 }

```

`\init@restore@glb@settings` This macros does by default nothing but get redefined inside `\check@mathfonts` to initiate fontsize restoring in nested formulas.

```

224 (-trace)\let\init@restore@glb@settings\relax
225 (*trace)
226 \def\init@restore@glb@settings{%
227   \ifnum \tracingfonts>\thr@@
228     \@font@info{*** MATH: no resetting (not in
229       nested math)}\fi
230 }
231 (/trace)

```

`\restglb@settings` This macro will be executed the first time after the current formula.

```

232 \def\restglb@settings{%
233   (*trace)
234     \ifnum \tracingfonts>\thr@@
235       \@font@info{*** MATH: restoring}\fi
236   (/trace)
237     \begingroup
238       \let\f@size\curr@math@size
239       \ifx\glb@currsiz \f@size
240   (*trace)
241     \ifnum \tracingfonts>\thr@@
242       \@font@info{*** MATH: ... already okay (\f@size)}\fi
243   (/trace)
244     \else
245   (*trace)
246     \ifnum \tracingfonts>\thr@@
247       \@font@info{*** MATH: ... to \f@size}\fi
248   (/trace)

```

```

249         \glb@settings
250         \fi
251     \endgroup
252 }

```

35.2.3 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 a 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 \mathcal{S} -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.⁶ 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 </2ekernel | package>

```

⁶One 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.

36 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 (*2ekernel)
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\@nil{%
305 \if>#2>\set@size@func@args#1\@nil
306 \let\sizefn@info\@empty
307 \else\expandafter\set@size@func@args\remove@star#2\@nil
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 #4 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 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` `\upper@bound` We use two `dimen` registers `\lower@bound` and `\upper@bound` to store the lower 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 size against `\f@size`. If it is not equal `\f@size` it is no good and we have to recurse.

```
356 \ifdim \f@size \p@=#1\p@
```

Otherwise if this test is passed we can try executing the size function.

```
357 \set@simple@size@args#2<#3\@nnil
```

```
358 \execute@size@function\sizefn@info
```

If the function was successful it should have left an external font name in `\external@font`. We use this to see if we can stop scanning. Otherwise we recurse.

```
359 \ifx\external@font\@empty
```

```
360 \else
```

```
361 \let\reserved@f\@empty
```

```
362 \fi
```

```
363 \fi
```

```
364 \reserved@f}
```

`\set@size@funct@args` This macro sets the optional and mandatory arguments for a size function. If the optional argument is not present it is set to the empty token list. The mandatory argument is delimited by the token `\@nil`.

```
365 \def\set@size@funct@args{\@ifnextchar[%
```

```
366 \set@size@funct@args@{\set@size@funct@args@[]}]}
```

```
367 \def\set@size@funct@args@[#1]#2\@nil{%
```

```
368 \def\mandatory@arg{#2}%
```

```
369 \def\optional@arg{#1}}
```

```
370 </2kernel)
```

`\DeclareSizeFunction` This function defines a new size function hiding the internal from the designer. The body of the size function may use `\optional@arg` and `\mandatory@arg` denoting the optional and mandatory argument that may follow the size specification `<...>`.

```
371 (*2kernel)
```

```
372 \def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
```

```
373 \@onlypreamble\DeclareSizeFunction
```

```
374 </2kernel)
```

`\execute@size@function` This macro is very simple. The only point worth noting is that calling an undefined size function will do nothing (actually execute a `\relax`).

```
375 (*2kernel | package)
```

```
376 \def\execute@size@function#1{%
```

```
377 (*trace)
```

```
378 \ifundefined{s@fct@#1}%
```

```
379 {\errmessage{Undefined font size function #1}}%
```

```

380         \s@fct@}%
381         {\csname s@fct@#1\endcsname}%
382 \trace)
383 (-trace)     \csname s@fct@#1\endcsname
384 }
385 \endkernel | package)

```

\try@size@range This macro tries to find a suitable range for requested size (specified by `\f@size`) in `\font@info`. All the relevant action is done in `\extract@rangefontinfo`. All that needs to be done is to stuff in the token list in `\font@info` so that `\extract@rangefontinfo` can inspect it. Note the `<-*\@nil>` token at the end to stop scanning.

```

386 (*2kernel)
387 \def\try@size@range{%
388     \expandafter\extract@rangefontinfo\font@info <-*\@nil<\@nnil
389 }

```

\try@size@substitution This is the last thing that can be tried. If the desired `\f@size` is found neither among the simple size specifications nor in one of the ranges the whole list of size specifications is searched for a nearby simple size.

```

390 \gdef\try@size@substitution{%

```

First we do some initializations. `\@tempdimb` will hold the difference between the wanted size and the best solution found so far, so we initialise it with `\maxdimen`. The macro `\best@size` will hold the best size found, nothing found is indicated by the empty value.

```

391     \@tempdimb \maxdimen
392     \let \best@size \@empty

```

Now we loop over the specification

```

393     \expandafter \try@simples \font@info <\number\@M>\@nil<\@nnil
394 }

```

\font@submax The macro `\font@submax` records the maximal deviation from the desired size encountered so far. Its value is used in a warning message at `\end{document}`. The macro `\fontsubfuzz` contains the amount that will not cause terminal warnings (warnings still go into the transcript file).

```

395 \def\font@submax{0pt}
396 \def\fontsubfuzz{.4pt}
397 \endkernel)
398 (+package)\def\fontsubfuzz{0pt}

```

\try@simples `\try@simples` goes through a font shape definition in the input until it recognizes the tokens `<*\@nil>`. It looks for simple sizes to determine the two closest sizes. It is assumed that simple sizes are in increasing order.

```

399 (*2kernel)
400 \gdef\try@simples#1<#2>{%
401     \tryif@simple#2->\tryif@simple}

```

\tryis@simple `\tryis@simple` is similar to `\is@range`. If it sees a simple size, it checks it against the value of `\f@size` and sets `\lower@font@size` or `\higher@font@size`. In the latter case, it stops the iteration. By adding `<\number\@M>` at the end of the line we always have an end point. This is a hack which probably should be corrected.

First it checks whether it is finished already, then whether the size specification in question is a simple one.

```
402 \gdef\tryif@simple#1-#2\tryif@simple{%
```

Most common case for `\reserved@f` first:

```
403 \let \reserved@f \try@simples
404 \if>#2%
```

If so, it compares it to the value of `\f@size`. This is done using a `dimen` register since there may be fractional numbers.

```
405 \dimen@ #1\p@
406 \ifdim \dimen@<\@M\p@
```

If `\dimen@` is `\@M\p@` we have reached the end of the fontspec (hopefully) otherwise we compare the value with `\f@size` and compute in `\@tempdimc` the absolute value of the difference between the two values.

```
407 \ifdim \f@size\p@<\dimen@
408 \@tempdimc \dimen@
409 \advance\@tempdimc -\f@size\p@
410 \else
411 \@tempdimc \f@size\p@
412 \advance\@tempdimc -\dimen@
413 \fi
```

The result is then compared with the smallest difference we have encountered, if the new value (in `\@tempdimc` is smaller) we have found a size which is a better approximation so we make it the `\best@size` and adjust `\@tempdimb`.

```
414 \ifdim \@tempdimc<\@tempdimb
415 \@tempdimb \@tempdimc
416 \def \best@size{#1}%
417 \fi
```

When we have reached the end of the fontspec we substitute the best size found (if any). We code this inline to save macro space; in the past this was done by a macro called `\subst@size`.

```
418 \else
```

`\subst@size` This macro substitutes the size recorded in `\best@size` for the unavailable size `\f@size`. `\font@submax` records the maximum difference between desired size and selected size in the whole run.

```
419 % \subst@size %% coded inline
420 % \def\subst@size{%
421 \ifx \external@font\@empty
422 \ifx \best@size\@empty
423 \else
424 \ifdim \@tempdimb>\font@submax \relax
425 \xdef \font@submax {\the\@tempdimb}%
426 \fi
427 \let \f@user@size \f@size
428 \let \f@size \best@size
429 \ifdim \@tempdimb>\fontsubfuzz\relax
430 \@font@warning{Font\space shape\space
431 'curr@fontshape'\space in\space size\space
432 <\f@user@size>\space not\space available\MessageBreak
433 size\space <\f@size>\space substituted}%
```

```

434     \fi
435     \try@simple@size
436     \do@subst@correction
437     \fi
438 \fi
439 % %}

```

This brings us back into the main part of `\tryif@simple`. Finally we get rid of any rubbish left over on the input stack.

```

440     \let \reserved@f \remove@to@nnil
441     \fi
442 \fi

```

If it's a range iterate also.

```

443 \reserved@f}

```

36.1 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@sfcnt@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@@
109 \gdef\newmathalphabet{%
\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.

37 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 ** is either `\reserved@e` (if the combination is not defined yet) or

```
{\family}{\series}{\shape}}
```

`\version@list` We initialize the version list to be empty.

```

16 \let\version@list=\@empty
17 \@onlypreamble\version@list

```

`\version@elt`

```

18 \let\version@elt\relax
19 \@onlypreamble\version@elt

```

`\new@mathversion` The macro `\new@mathversion` is called with the version control sequence as its argument.

```
20 %\def\new@mathversion#1{%
```

The first thing this macro does is to check if the version identifier is already present in `\version@list`. We enclose `\version@list` in braces since it might be empty (if no *version* is defined yet). But this means that we need a suitable number of `\expandafter` primitives.

```

21 % \expandafter\in@\expandafter#1\expandafter{\version@list}%
22 % \ifin@

```

If so it prints an error message. The `\next` macro is used to get rid of the four characters `\mv@` that would otherwise appear at the begin of the version name in the error message.

```

23 % \latex@error{Math version
24 % \expandafter@gobblefour\string#1'
25 % already defined}\@eha

```

Otherwise we have a new version, and we can proceed with entering it into the tables. We add it to `\version@list`. This is very easy: we define `\version@elt` (which is the delimiter in `\version@list`) to protect itself and the following token from being expanded and simply redefine `\version@list`.

```

26 % \else
27 % \global\expandafter\newcount\c\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 *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 *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 *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 *alphabet identifier* \rangle \langle *internal group number* \rangle \langle *default font assignments* \rangle ...

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 \beginingroup
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> \beginingroup
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 \ifmode
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> \ifmode
175 <latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\sixt@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\font@encoding{##1}\def\font@family{##2}%
208     \try@load@fontshape
209     \endgroup
210     \fi
211     \expandafter
212     \ifx\csname##1/##2/##3/##4\endcsname\relax
213       \@latex@error{This NFSS system isn't set up properly}%
214       {For encoding scheme ##1 the defaults
215         ##2/##3/##4 do not form a valid font shape}%
216     \else
217       \@font@info{... okay}%
218     \fi}%
219   \cdp@list

```

Now we make sure that `\error@fontshape` is okay.

```

220   \begingroup
221     \escapechar\m@ne
222     \error@fontshape
223     \expandafter\ifx\csname \curr@fontshape\endcsname\relax
224       \begingroup
225         \try@load@fontshape
226       \endgroup
227     \fi
228     \expandafter\ifx\csname \curr@fontshape\endcsname\relax
229       \@latex@error{This NFSS system isn't set up properly}%
230       {The system maintainer forgot to specify a suitable
231         substitution
232         font shape using the \noexpand\DeclareErrorFont
233         command}%
234     \fi
235   \endgroup

```

Set `\select@group` to its meaning used within the document body.

```

236   \let\select@group\document@select@group

```

Install the default font attributes they are currently pointing to error font shape. Don't use `\reset@font` since that would trigger `\selectfont`.

```

237   \fontencoding{\encodingdefault}%

```

```

238 \fontfamily{\familydefault}%
239 \fontseries{\seriesdefault}%
240 \fontshape{\shapedefault}%
kill all macros not longer needed. we need to add many more!!!!!!
241 \everyjob{}%
242 }
243 \@onlypreamble\process@table

244 %\@onlypreamble\set@mathradical

```

\DeclareMathVersion

```

245 \def\DeclareMathVersion#1{%
246 \expandafter\new@mathversion\csname mv@#1\endcsname}
247 \@onlypreamble\DeclareMathVersion

```

\new@mathversion

```

248 \def\new@mathversion#1{%
249 \expandafter\in@\expandafter#1\expandafter{\version@list}%
250 \ifin@
251 \@font@info{Redefining math version
252 \expandafter\@gobblefour\string#1}%
253 \else
254 \expandafter\newcount\csname c@\expandafter
255 \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   \@tempwafalse
282   \edef\reserved@b{#2}%
283   \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
284     \ifx\reserved@b\reserved@c \@tempwattrue\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 \@tempswatruel\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 \@tempwatruel\fi}%
485   \cdp@list
486   \if@tempswa
487     \expandafter\SetMathAlphabet@
488     \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
489     \endcsname \csname M@#3\expandafter\endcsname
490     \csname \expandafter\@gobble\string#1\space\endcsname#1%
491   \else
492     \@latex@error{Encoding scheme ‘#3’ unknown}\@eha
493   \fi
494 }
495 \@onlypreamble\SetMathAlphabet

```

\SetMathAlphabet@

```

496 \def\SetMathAlphabet@#1#2#3#4#5{%
497   \expandafter\in@\expandafter#1\expandafter{\version@list}%
498   \ifin@
499     \expandafter\in@\expandafter#4\expandafter{\alpha@list}%
500   \ifin@
501     \begingroup
502     \toks@{}%
503     \def\getanddefine@fonts##1##2{%
504       \addto@hook\toks@{\getanddefine@fonts##1##2}%
505     }%
506     \def\reserved@c##1##2##3##4{%

```

% for message below

```

507         \expandafter\@gobble\string##4}%
508     \def\install@mathalphabet##1##2{%
509         \ifx##1#4%
510             \addto@hook\toks@
511                 {\install@mathalphabet#4{\select@group#4#3#2}}%
512             \@font@info{Overwriting math alphabet
513                 ‘\string#5’ in version ‘\expandafter
514                 \@gobblefour\string#1’\MessageBreak
515                 \spaces \reserved@c##2 -->
516                 \expandafter\@gobble\string#2}%
517         \else
518             \addto@hook\toks@{\install@mathalphabet##1{##2}}%
519         \fi
520     }%
521     #1%
522     \xdef#1{\the\toks@}%
523 \endgroup
524 \else

```

If the math alphabet was defined via `\DeclareSymbolFontAlphabet` we have remove its external definition and add it as a normal math alphabet to every version before trying to change it in one version.

```

525     \edef\reserved@a{%
526         \noexpand\in@{\string\use@mathgroup}{\meaning#4}}%
527     \reserved@a
528     \ifin@
529         \def\reserved@b##1\use@mathgroup##2##3{%
530             \def\reserved@b{##3}\def\reserved@c{##2}}%
531         \expandafter\reserved@b#4%
532     \begingroup
533         \def\install@mathalphabet##1##2{%
534             \addto@hook\toks@{\install@mathalphabet##1{##2}}%
535         }%
536         \def\getanddefine@fonts##1##2{%
537             \addto@hook\toks@{\getanddefine@fonts##1##2}%
538             \ifnum##1=\reserved@b
539                 \expandafter
540                 \addto@hook\expandafter\toks@
541                 \expandafter{\expandafter\install@mathalphabet
542                 \expandafter#4\expandafter
543                 {\expandafter\select@group\expandafter
544                 #4\reserved@c##2}}%
545             \fi
546         }%
547         \def\version@elt##1{%
548             \toks@{}%
549             ##1%
550             \xdef##1{\the\toks@}%
551         }%
552         \version@list
553     \endgroup

```

Put it into the `\alpha@list` with default ‘error’

```

554         \expandafter\gdef\expandafter\alpha@list\expandafter
555         {\alpha@list

```

```

556         \alpha@elt #4\no@alphabet@error \no@alphabet@error}%
557         \gdef#4{\no@alphabet@error #5}% fake things :-)

```

Then call the internal setting routine again:

```

558         \SetMathAlphabet@{#1}{#2}{#3}#4#5%
559     \else
560         \@latex@error{Command ‘\string#5’ not defined as a
561             math alphabet}%
562         {Use \noexpand\DeclareMathAlphabet to define it.}%
563     \fi
564 \fi
565 \else
566     \@latex@error{Math version ‘\expandafter@gobblefour\string#1’
567         is not
568         defined}{You probably misspelled the name of the math
569         version.^^JOr you have to specify an additional package.}%
570 \fi
571 }
572 \@onlypreamble\SetMathAlphabet@

```

`\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\expandafter\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> \@latex@error{Symbol font ‘#3’ is not defined}\@eha
662 <latexrelease> \fi
663 <latexrelease>}
664 <latexrelease>\EndIncludeInRelease
665 <*2ekernel>

666 \@onlypreamble\DeclareMathAccent

```

`\set@mathaccent`

```

667 </2ekernel>
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 </2ekernel | 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 \begin@group
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\reservedb\@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{%
759   7 is the default value returned in the case that \mathchar@type is passed some-
760   thing unexpected, like a math symbol font name. We locally move \mathalpha
761   out of the way so if you use that the right branch is taken. This will still fail if an
762   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 \def\@DeclareMathDelimiter#1#2#3#4#5#6{%
772   \expandafter\in@\csname sym#3\expandafter\endcsname
773   \expandafter{\group@list}%
774   \ifin@
775     \expandafter\in@\csname sym#5\expandafter\endcsname
776     \expandafter{\group@list}%
777   \ifin@
778     \begingroup
779       \count\z@=#4\relax
780       \count\tw@\count\z@
781       \divide\count\z@\sixt@n
782       \count@\count\z@
783       \multiply\count@\sixt@n

```

```

784     \advance\count\tw@-\count@
785     \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
786     %
787     \count\z@=#6\relax
788     \count\tw@\count\z@
789     \divide\count\z@\sist@n
790     \count@\count\z@
791     \multiply\count@\sist@n
792     \advance\count\tw@-\count@
793     \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
794     %
795     \edef\reserved@a{\noexpand\in@
796       {\expandafter\@gobble\string\delimiter}{\meaning#1}}%
797     \reserved@a
798     \ifin@
799       \expandafter\set@mathdelimiter
800       \csname sym#3\expandafter\endcsname
801       \csname sym#5\endcsname#1#2%
802       \reserved@c\reserved@d
803       \@font@info{Redeclaring math delimiter \string#1}%
804     \else
805       \expandafter\ifx
806       \csname\expandafter\@gobble\string#1\endcsname
807       \relax
808       \expandafter\set@mathdelimiter
809       \csname sym#3\expandafter\endcsname
810       \csname sym#5\endcsname#1#2%
811       \reserved@c\reserved@d
812     \else
813       \@latex@error{Command '\string#1' already defined}\@eha
814     \fi
815   \fi
816 \endgroup
817 \else
818   \@latex@error{Symbol font '#5' is not defined}\@eha
819 \fi
820 \else
821   \@latex@error{Symbol font '#3' is not defined}\@eha
822 \fi
823 }
824 \@onlypreamble\@DeclareMathDelimiter

```

\@xDeclareMathDelimiter

```

825 \def\@xDeclareMathDelimiter#1#2#3#4#5{%
826   \expandafter\in@\csname sym#2\expandafter\endcsname
827   \expandafter{\group@list}%
828   \ifin@
829     \expandafter\in@\csname sym#4\expandafter\endcsname
830     \expandafter{\group@list}%
831   \ifin@
832     \begin@group
833     \count\z@=#3\relax
834     \count\tw@\count\z@
835     \divide\count\z@\sist@n

```

```

836     \count@\count\z@
837     \multiply\count@\sist@0n
838     \advance\count\tw@-\count@
839     \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
840     %
841     \count\z@=#5\relax
842     \count\tw@\count\z@
843     \divide\count\z@\sist@0n
844     \count@\count\z@
845     \multiply\count@\sist@0n
846     \advance\count\tw@-\count@
847     \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
848     \expandafter\set@mathdelimiter
849     \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
850     \reserved@c\reserved@d
851     \endgroup
852     \else
853     \@latex@error{Symbol font ‘#4’ is not defined}\@eha
854     \fi
855     \else
856     \@latex@error{Symbol font ‘#2’ is not defined}\@eha
857     \fi
858 }
859 \@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)

```

860 </2kernel>
861 (*2kernel | latexrelease)
862 (latexrelease)\IncludeInRelease{2019/10/01}%
863 (latexrelease)           {\set@mathdelimiter}{make delimiters robust}%
864 \def\set@mathdelimiter#1#2#3#4#5#6{%

```

We use `\protected` not `\MakeRobust` so that `\bigl\lfloor` etc. works inside `\protected@edef`.

```

865     \protected
866     \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
867     \hexnumber@#2#6 }%
868     % \MakeRobust#3%
869 }
870 \@onlypreamble\set@mathdelimiter
871 </2kernel | latexrelease>
872 (latexrelease)\EndIncludeInRelease
873 (latexrelease)\IncludeInRelease{0000/00/00}%
874 (latexrelease)           {\set@mathdelimiter}{make delimiters robust}%
875 (latexrelease)
876 (latexrelease)\def\set@mathdelimiter#1#2#3#4#5#6{%
877 (latexrelease) \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
878 (latexrelease) \hexnumber@#2#6 }}
879 (latexrelease)
880 (latexrelease)\EndIncludeInRelease
881 (*2kernel)

```

```

\set@@mathdelimiter
882 \def\set@@mathdelimiter#1#2#3#4#5{%
883   \global\delcode'#3="\hexnumber@#1#4\hexnumber@#2#5\relax}
884 \@onlypreamble\set@@mathdelimiter

\DeclareMathRadical
885 \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!
886   \expandafter\ifx
887     \csname\expandafter\@gobble\string#1\endcsname
888     \relax
889     \let#1\radical
890   \fi
891   \edef\reserved@a{\noexpand\in@
892     {\expandafter\@gobble\string\radical}{\meaning#1}}%
893   \reserved@a
894   \ifin@
895     \expandafter\in@\csname sym#2\expandafter\endcsname
896     \expandafter{\group@list}%
897   \ifin@
898     \expandafter\in@\csname sym#4\expandafter\endcsname
899     \expandafter{\group@list}%
900   \ifin@
901     \begingroup
902       \count\z@=#3\relax
903       \count\tw@\count\z@
904       \divide\count\z@\sist@@n
905       \count@\count\z@
906       \multiply\count@\sist@@n
907       \advance\count\tw@-\count@
908       \edef\reserved@c{%
909         \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
910       \count\z@=#5\relax
911       \count\tw@\count\z@
912       \divide\count\z@\sist@@n
913       \count@\count\z@
914       \multiply\count@\sist@@n
915       \advance\count\tw@-\count@
916       \edef\reserved@d{%
917         \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
Coded inline instead of using \set@mathradical
918 %       \expandafter\set@mathradical
919 %       \csname sym#2\expandafter\endcsname
920 %       \csname sym#4\endcsname#1%
921 %       \reserved@c\reserved@d
922 %       \xdef#1{\radical"\expandafter\hexnumber@
923 %           \csname sym#2\endcsname\reserved@c
924 %           \expandafter\hexnumber@
925 %           \csname sym#4\endcsname\reserved@d
926 %           \relax}%
927 %       \endgroup

```

```

928     \else
929       \@latex@error{Symbol font ‘#4’ is not defined}\@eha
930     \fi
931   \else
932     \@latex@error{Symbol font ‘#2’ is not defined}\@eha
933   \fi
934 \else
935   \@latex@error{Command ‘\string#1’ already defined}\@eha
936 \fi
937 }
938 \@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
939 \let\mathalpha\relax

```

`\mathchar@type`

```

940 \def\mathchar@type#1{%
941   \ifodd 2#1#1 #1\else           % is this non-negative number?
942     \ifx#1\mathord 0\else
943     \ifx#1\mathop 1\else
944     \ifx#1\mathbin 2\else
945     \ifx#1\mathrel 3\else
946     \ifx#1\mathopen 4\else
947     \ifx#1\mathclose 5\else
948     \ifx#1\mathpunct 6\else
949     7%           % anything else is variable ord
950     \fi
951     \fi
952     \fi
953     \fi
954   \fi
955   \fi
956   \fi
957 \fi}
958 \@onlypreamble\mathchar@type

```

`\DeclareSymbolFontAlphabet`

```

959 \def\DeclareSymbolFontAlphabet#1#2{%
960   \expandafter\DeclareSymbolFontAlphabet@
961   \csname \expandafter@gobble\string#1\space\endcsname{#2}#1}
962 \@onlypreamble\DeclareSymbolFontAlphabet

```

`\DeclareSymbolFontAlphabet@`

```

963 \def\DeclareSymbolFontAlphabet@#1#2#3{%
We use the switch \if@tempswa to decide if we can declare this symbol font
alphabet.
964   \@tempswatrue

```

First check if #2 is known to be a symbol font

```
965 \expandafter\in@\csname sym#2\expandafter\endcsname
966 \expandafter{\group@list}%
967 \ifin@
```

Check if #1 is defined as a math alphabet defined via `\DeclareMathAlphabet`:

```
968 \expandafter\in@\expandafter#1\expandafter{\alpha@list}%
969 \ifin@
```

If so remove it from the `\alpha@list` and from all math version macros.

```
970 \@font@info{Redeclaring math alphabet \string#3}%
971 \toks@{}%
972 \def\alpha@elt##1##2##3{%
973 \ifx##1#1\else\addto@hook\toks@{\alpha@elt##1##2##3}\fi}%
974 \alpha@list
975 \xdef\alpha@list{\the\toks@}%
```

Now we loop over all versions and remove the math alphabet:

```
976 \def\version@elt##1{%
977 \begingroup
978 \toks@{}%
979 \def\getanddefine@fonts####1####2{%
980 \addto@hook\toks@{\getanddefine@fonts####1####2}}%
981 \def\install@mathalphabet####1####2{%
982 \ifx####1#1\else
983 \addto@hook\toks@{\install@mathalphabet
984 ####1{####2}}\fi}%
985 ##1%
986 \xdef##1{\the\toks@}%
987 \endgroup
988 }%
989 \version@list
990 \else
```

If #3 is not defined as a math alphabet check if it is defined at all:

```
991 \expandafter\ifx
992 \csname\expandafter\@gobble\string#1\space\endcsname
993 \relax
```

If it is undefined, fine otherwise check if it is a math alphabet defined via `\DeclareSymbolFontAlphabet`:

```
994 \else
995 \edef\reserved@a{%
996 \noexpand\in@{\string\use@mathgroup}{\meaning#1}}%
997 \reserved@a
998 \ifin@
999 \@font@info{Redeclaring math alphabet \string#3}%
1000 \else
```

Since the command #3 is defined to be something which is not a math alphabet we have to skip redefining it.

```
1001 \@tempwafalse
1002 \@latex@error{Command ‘\string#3’ already defined}\@eha
1003 \fi
1004 \fi
1005 \fi
1006 \else
```

Since the symbol font is not known we better skip defining this alphabet.

```

1007     \@tempswafalse
1008     \@latex@error{Unknown symbol font '#2'}\@eha
1009     \fi
1010     \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.

```

1011     \def\group@elt##1##2{%
1012         \expandafter\ifx\csname sym#2\endcsname##1%
1013         \expandafter\reserved@a\string##2\@nil
1014         \fi}%
1015     \def\reserved@a##1##2/##3\@nil{%
1016         \def\reserved@a{##2}}%
1017     \group@list
1018     \toks@{\relax\ifmmode \else \non@alpherr#1\fi}%
1019     \edef#1{\the\toks@
1020         \noexpand\use@mathgroup
1021         \expandafter\noexpand\csname M@\reserved@a\endcsname
1022         \csname sym#2\endcsname}%
1023     \def#3{\protect#1}%
1024     \fi
1025 }
1026 \@onlypreamble\DeclareSymbolFontAlphabet@
1027 /2kernel)

```

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.

38 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 <*2ekernel>
```

38.1 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\ttddefault\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}

```

39 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 </2ekernel)
32 <*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!⁷

```
\DeclareFontSeriesDefault The command to declare font series defaults for the "rm", "sf" or "tt" family.
35 \newcommand\DeclareFontSeriesDefault[3][]{%
36   \def\reserved@a{#1}%
No optional argument: set up general default.
37   \ifx\reserved@a\@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 argument must be 'rm', 'sf', or 'tt'. \MessageBreak
56     Mandatory first 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 `\mdseries@sf` `\DeclareFontSeriesDefault` (or possibly directly).

`\mdseries@tt` 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.

`\bfseries@rm` To account for the fact that by default we typeset in CM or LM we set up the `\bfseries@..` defaults to use `bx` instead.

`\bfseries@sf` 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.

⁷I 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.

```

60 \def\bfseries@rm{bx}
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\sfdef@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 the select the correct family default in that case.

```

79 \expand@font@defaults
80 \ifx\f@family\rmdef@ult \fontseries\bfseries@rm
81 \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
82 \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt

```

If not `\bfdefault` is used.

```

83 \else \fontseries\bfdefault
84 \fi\fi\fi
85 \selectfont
86 }

```

`\mdseries` This document command switches to the medium series.

```

87 \DeclareRobustCommand\mdseries{%
88 \not@math@alphabet\mdseries\relax
89 \expand@font@defaults

```

```

90   \ifx\f@family\rmdef@ult      \fontseries\mdseries@rm
91   \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
92   \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
93   \else                        \fontseries\mddefault
94   \fi\fi\fi
95   \selectfont
96 }

```

`\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 stream-lined 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\mdef@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\mdef@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`.

```

97 \DeclareRobustCommand\rmfamily{%
98   \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.

```
99   \prepare@family@series@update{rm}\rmdefault
```

Then comes the hook code (by default a no-op) and finally the call to `\selectfont`.

```

100   \@rmfamilyhook
101   \selectfont}

```

`\sffamily` `\ttfamily` The definitions for `\sffamily` and `\ttfamily` are similar, the differences are only in what font families get checked.

```

102 \DeclareRobustCommand\sffamily{%
103   \not@math@alphabet\sffamily\mathsf

```

```

104 \prepare@family@series@update{sf}\sfdefault
105 \@sffamilyhook
106 \selectfont}

107 \DeclareRobustCommand\ttfamily{%
108 \not@math@alphabet\ttfamily\mathtt
109 \prepare@family@series@update{tt}\ttdefault
110 \@ttfamilyhook
111 \selectfont}

\@rmfamilyhook By default the hooks do nothing.
\@sffamilyhook 112 \let\@rmfamilyhook\@empty
\@ttfamilyhook 113 \let\@sffamilyhook\@empty
114 \let\@ttfamilyhook\@empty

\series@change@debug For debugging, but right now none of this code is extracted. The idea is to have
a separate package with debugging code one day.
115 (*debug)
116 \let\series@change@debug\typeout
117 \let\series@change@debug@gobble
118 </debug>

\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 \if@forced@series
122 (+debug) \series@change@debug{No series preparation (forced \f@series)\on@line}%
123 \fontfamily#2%
124 \else
125 (+debug) \series@change@debug{Preparing for switching to #1 (#2)\on@line}%
126 \expand@font@defaults

We prepare for changing the current series. We have to find it before changing
the family as discussed above.
127 \let\target@series@value\@empty
128 \def\target@meta@family@value{#1}%

As the very last item in the meta family list we add \@elt{??} and define this
pseudo meta family to be the current font family. So if none of the real meta
families matched then this will match. This will cover the following case:

```

- \bfseries is called for a family using bx (e.g., CMR)
- Switch to a font family that is none of the the meta families, e.g., via \fontfamily{ptm}\selectfont
- Then none of the real meta families, match but the final \@elt{??} will.

- Therefore if the current series is `\mddefault` or `\bfdefault` it will be detected and the corresponding target series selected.

```
129 \expandafter\edef\csname ??def@ult\endcsname{\f@family}%
```

To find it we loop over the meta family list with a suitable definition of `\@elt`.

```
130 \let\@elt\update@series@target@value
131 \meta@family@list
```

Last resort pseudo meta family. Will only be looked at if none of the real ones have matched.

```
132 \@elt{??}%
133 \let\@elt\relax
```

That will figure out the correct series value to use without updating it. Now we can change the family.

```
134 \fontfamily#2%
```

After that we update the series. That code is again like the one above.

```
135 \ifx\target@series@value\empty
136 (+debug) \series@change@debug{Target series still empty ...}%
137 \else
138 \ifx \f@series\target@series@value
139 (+debug) \series@change@debug{Target series unchanged:
140 (+debug) \f@series \space = \target@series@value}%
141 \else
142 \maybe@load@fontshape
143 (+debug) \series@change@debug{Target series:
144 (+debug) \f@series \space -> \target@series@value}%
145 \let\f@series\target@series@value
146 \fi
147 \fi
148 \fi
149 }
```

`\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.

```
150 \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.

```
151 \def\reserved@a{#1}%
152 \ifx\target@meta@family@value\reserved@a % rm -> rm do nothing
153 \else
154 (+debug) \series@change@debug{Trying to match #1: \csname#1def@ult\endcsname
155 (+debug) \space = \f@family\space ?}%
```

We only “do” something if the current font family matches the current meta family.

```
156 \expandafter\ifx\csname#1def@ult\endcsname\f@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.

```
157 \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.

```

158     \expandafter\let\expandafter\reserved@b
159             \csname mdseries@\target@meta@family@value\endcsname
160     \expandafter\let\expandafter\reserved@c
161             \csname bfseries@\target@meta@family@value\endcsname
162 (+debug)\series@change@debug{Targets for mdseries and bfseries:
163 (+debug)             \reserved@b\space and \reserved@c}%

```

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.

```

164     \expandafter\ifx\csname mdseries@#1\endcsname\f@series
165 (+debug) \series@change@debug{mdseries@#1 matched -> \reserved@b}%
166             \let\target@series@value\reserved@b
167     \else\expandafter\ifx\csname bfseries@#1\endcsname\f@series
168 (+debug) \series@change@debug{bfseries@#1 matched -> \reserved@c}%
169             \let\target@series@value\reserved@c
170     \else\ifx\f@series\mdef@ult \let\target@series@value\reserved@b
171 (+debug) \series@change@debug{mdef@ult matched -> \reserved@b}%
172     \else\ifx\f@series\bfdef@ult \let\target@series@value\reserved@c
173 (+debug) \series@change@debug{bfdef@ult matched -> \reserved@c}%
174     \fi\fi\fi\fi
175     \fi
176     \fi
177 }

```

`\init@series@setup` This is code to be run at begin document ...

```

178 \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.

```

179 \ifx\bfseries@rm@kernel\bfseries@rm
180 \expandafter\in@\expandafter{\rmdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
181 \ifin@ \else \def\bfseries@rm{b}\fi\fi

```

Same approach for `\bfseries@sf` and `\bfseries@tt`:

```

182 \ifx\bfseries@sf@kernel\bfseries@sf
183 \expandafter\in@\expandafter{\sfdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
184 \ifin@ \else \def\bfseries@sf{b}\fi\fi
185 \ifx\bfseries@tt@kernel\bfseries@tt

```

```

186 \expandafter\in@\expandafter{\ttdefault}{cmr,cmss,cmtt,lcms,lcmtt,lmr,lmss,lmtt}%
187 \ifin@ \else \def\bfseries@tt{b}\fi\fi

```

If the document preamble has changed the `\familydefault` or 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.

```

188 \expand@font@defaults
189 \ifx\famdef@ult\rmdef@ult \rmfamily
190 \else\ifx\famdef@ult\sfdef@ult \sffamily
191 \else\ifx\famdef@ult\ttdef@ult \ttfamily
192 \fi\fi\fi
193 }%

```

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.

```

194 \expandafter\let\csname ver@mweights.sty\endcsname\fmtversion
195 </2kernel | latexrelease>
196 <latexrelease>\EndIncludeInRelease
197 <latexrelease>\IncludeInRelease{0000/00/00}%
198 <latexrelease> \DeclareFontSeriesDefault{Custom series}%
199 <latexrelease>
200 <latexrelease>\let\DeclareFontSeriesDefault\@undefined
201 <latexrelease>\let\bfseries@rm\@undefined
202 <latexrelease>\let\bfseries@sf\@undefined
203 <latexrelease>\let\bfseries@tt\@undefined
204 <latexrelease>\let\bfseries@rm@kernel\@undefined
205 <latexrelease>\let\bfseries@sf@kernel\@undefined
206 <latexrelease>\let\bfseries@tt@kernel\@undefined
207 <latexrelease>\let\mdseries@rm\@undefined
208 <latexrelease>\let\mdseries@sf\@undefined
209 <latexrelease>\let\mdseries@tt\@undefined
210 <latexrelease>\let\expand@font@defaults\@undefined
211 <latexrelease>\expandafter\let\csname ver@mweights.sty\endcsname\@undefined
212 <latexrelease>
213 <latexrelease>\DeclareRobustCommand\bfseries
214 <latexrelease> \not@math@alphabet\bfseries\mathbf
215 <latexrelease> \fontseries\bfdefault\selectfont}
216 <latexrelease>\DeclareRobustCommand\mdseries
217 <latexrelease> \not@math@alphabet\mdseries\relax
218 <latexrelease> \fontseries\mddefault\selectfont}
219 <latexrelease>\DeclareRobustCommand\rmfamily
220 <latexrelease> \not@math@alphabet\rmfamily\mathrm
221 <latexrelease> \fontfamily\rmdefault\selectfont}
222 <latexrelease>\DeclareRobustCommand\sffamily

```

```

223 <latexrelease>          {\not@math@alphabet\sffamily\mathsf
224 <latexrelease>          \fontfamily\sfdefault\selectfont}
225 <latexrelease>\DeclareRobustCommand\ttfamily
226 <latexrelease>          {\not@math@alphabet\ttfamily\mathtt
227 <latexrelease>          \fontfamily\ttdefault\selectfont}
228 <latexrelease>
229 <latexrelease>\let\@rmfamilyhook\@undefined
230 <latexrelease>\let\@sffamilyhook\@undefined
231 <latexrelease>\let\@ttfamilyhook\@undefined
232 <latexrelease>\let\@meta@family@list\@undefined
233 <latexrelease>\let\prepare@family@series@update\@undefined
234 <latexrelease>\let@update@series@target@value\@undefined
235 <latexrelease>

```

This is always called in `\document` so don't make it undefined.

```

236 <latexrelease>\let\init@series@setup\relax
237 <latexrelease>
238 <latexrelease>\EndIncludeInRelease
239 <*/2kernel>

```

40 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 surrious 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.⁸ Further nested calls restart at the beginning.

```

240 </2kernel>
241 <*/2kernel | latexrelease>

```

⁸Maybe 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 ...

```

242 (latexrelease)\IncludeInRelease{2020/02/02}%
243 (latexrelease)          {\DeclareEmphSequence}{Nested emph}%
244 \def\DeclareEmphSequence#1{%
245   \protected@edef\emfontdeclare@clist{\zap@space#1, \empty\emforce\emreset}%
246 }

```

By default the it is empty, in which case `\emminnershape` is used by L^AT_EX.

```
247 \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:

```
248 \DeclareRobustCommand\emreset{\upshape\ulcshape}
```

`\em` The new definition for `\em` (and implicitly `\emph` is like it was before if `\emfontdeclare@clist` is empty.

```

249 \DeclareRobustCommand\em{%
250   \@nomath\em
251   \ifx\emfontdeclare@clist\empty
252     \ifdim \fontdimen\@ne\font >\z@
253       \emminnershape \else \itshape \fi
254   \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.

```
255 \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.

```

256   \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
257   \fi
258 }

```

```
259 \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`.

```
260 \def\do@emfont@update#1,#2\do@emfont@update{%
```

First action is to alter the list and move the first entry to the end

```
261 \def\emfontdeclare@clist{#2,#1}%
```

Then we execute current declaration. Appending `\selectfont` means one can write just `\fontshape{it}` and that works then too.

```
262 % \typeout{Use: \detokenize{#1}}%
```

```
263 #1\selectfont
```

We then compare the current font with our saved version, but with a slight twist: we add `\em@force` at the end of the name. Normally this is empty so has no effect but if there was an `\emforce` as part of `#1` it will append a `/` to the font name (making it invalid) thus this will then always fail the test.

If the test fails we are done and the declarations will be used. Otherwise we will try the next declaration in the sequence.

```
264 \expandafter\ifx\csname \curr@fontshape/\f@size\em@force
```

For the comparison with `\ifx` we have to expand `\em@currfont` once as the relevant info is inside.

```

265             \expandafter\endcsname
266             \em@currfont
267 \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
If \emforce was used, we have to undo its effect:
268 \else
269   \let\em@force\@empty
270 \fi
271 }

```

`\emforce` The definition of `\emforce` is simple: change `\em@force` to make the above test always invalid.

```

272 \protected\def\emforce{\def\em@force{}}
273 \let\em@force\@empty
274 \ifx\em@force\@empty
275 \ifx\em@force\@empty

```

`\em` These are the older definitions for `\em`, prior to 2020.

`\em` We also have to define the *emphasize* font change command (i.e. `\em`). This command will look is the current font is sloped (i.e. has a positive `\fontdimen1`) and will then select either `\upshape` or `\itshape`.

```

276 \ifx\em@force\@empty
277 \ifx\em@force\@empty
278 \ifx\em@force\@empty
279 \ifx\em@force\@empty
280 \ifx\em@force\@empty
281 \ifx\em@force\@empty
282 \ifx\em@force\@empty
283 \ifx\em@force\@empty
284 \ifx\em@force\@empty
285 \ifx\em@force\@empty
286 \ifx\em@force\@empty
287 \ifx\em@force\@empty
288 \ifx\em@force\@empty
289 \ifx\em@force\@empty
290 \ifx\em@force\@empty
291 \ifx\em@force\@empty
292 \ifx\em@force\@empty
293 \ifx\em@force\@empty
294 \ifx\em@force\@empty
295 \ifx\em@force\@empty

```

`\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`.

```

296 \def\not@math@alphabet#1#2{%
297   \relax
298   \ifmmode
299     \@latex@error{Command \noexpand#1invalid in math mode}%
300     {%
301       Please
302       \ifx#2\relax

```

```

303         define a new math alphabet^^J%
304         if you want to use a special font in math mode%
305         \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 :-).

```

306         use the math alphabet \noexpand#2instead of
307         the #1command%
308         \fi
309         .
310     }%
311 \fi}

```

Finally we provide two abbreviations to switch to the L^AT_EX *versions*.

```

312 \DeclareRobustCommand\boldmath{\@nomath\boldmath
313         \mathversion{bold}}
314 \DeclareRobustCommand\unboldmath{\@nomath\unboldmath
315         \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`.

```

316 \def\math@version{normal}

```

40.1 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.

```

317 \def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
318 \DeclareRobustCommand\symbol[1]{\char #1\relax}

```

`\@setfontsize` This abbreviation is used by L^AT_EX's user level size changing commands, such as
`\@setsize` `\large`.

```

319 \def\@setfontsize#1#2#3{\@nomath#1%

```

For the benefit of people relying on keeping the name of the current font command saved in `\@currsize` we define it. To ensure that `\@setfontsize` keeps being robust we omit this assignment during times where `\protect` differs from `\@typeset@protect`.

```

320     \ifx\protect\@typeset@protect
321         \let\@currsize#1%
322     \fi
323     \fontsize{#2}{#3}\selectfont}

```

For compatibility we also define `\@setsize` the 209 command

```

324 \*compat)
325 \def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
326 \</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`.

```

327 \def\hexnumber@#1{\ifcase\number#1
328 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
329 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.

```

330 \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`.

```

331 %\DeclareRobustCommand\copyright
332 %   {\ooalign{\hfil
333 %     \raise.07ex\hbox{\mdseries\upshape c}\hfil\cr
334 %     \mathhexbox20D}}

```

`\normalfont` The macro `\reset@font` is used in \LaTeX to switch to a standard font, in order to initialize the current font in situations where typesetting is done in a new visual context (e.g. in a footnote). We define it here to allow the test for the new \LaTeX version above but nevertheless are able to run all kind of mixtures.

The user interface name for `\reset@font` is `\normalfont`:

```

335 </2ekernel)
336 (*2ekernel | latexrelease)
337 (latexrelease)\IncludeInRelease{2020/02/02}%
338 (latexrelease)           {\normalfont}{Add hook to \normalfont}%
339 \DeclareRobustCommand\normalfont{%

```

Instead of calling `\usefont`, as it was done in the past, we inline the code from `\usefont` as we want to add the hook before `\selectfont`, but after all the font attributes are set.

```

340   \fontencoding\encodingdefault
341   \edef\f@family{\familydefault}%
342   \edef\f@series{\seriesdefault}%
343   \edef\f@shape{\shapedefault}%
344   \@defaultfamilyhook
345   \selectfont}
346 \let\reset@font\normalfont

```

`\@defaultfamilyhook` By default the hooks do nothing.

```

347 \let\@defaultfamilyhook\@empty
348 </2ekernel | latexrelease)
349 (latexrelease)\EndIncludeInRelease
350 (latexrelease)\IncludeInRelease{0000/00/00}%

```

```

351 (latexrelease)          {\normalfont}{Add hook to \normalfont}%
352 (latexrelease)
353 (latexrelease)\DeclareRobustCommand\normalfont
354 (latexrelease)          {\usefont\encodingdefault
355 (latexrelease)          \familydefault
356 (latexrelease)          \seriesdefault
357 (latexrelease)          \shapedefault
358 (latexrelease)          \relax}
359 (latexrelease)\let\reset@font\normalfont
360 (latexrelease)
361 (latexrelease)\let\@defaultfamilyhook\@undefined
362 (latexrelease)
363 (latexrelease)\EndIncludeInRelease
364 (*2ekernel)

```

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.

```

365 \def\not@base#1{\@latex@error
366   {Command \noexpand#1not provided in base LaTeX2e}%
367   {Load the latexsym or the amsfonts package to
368   define this symbol}}
369 \def\mho{\not@base\mho}
370 \def\Join{\not@base\Join}
371 \def\Box{\not@base\Box}
372 \def\Diamond{\not@base\Diamond}
373 \def\leadsto{\not@base\leadsto}
374 \def\squsubset{\not@base\squsubset}
375 \def\sqsupset{\not@base\sqsupset}
376 \def\lhd{\not@base\lhd}
377 \def\unlhd{\not@base\unlhd}
378 \def\rhd{\not@base\rhd}
379 \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.

```

380 \DeclareErrorFont{OT1}{cmr}{m}{n}{10} %% don't modify this setting
381                                         %% overwrite it in fontdef.cfg
382                                         %% 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`.

```

383 \fontfamily{cmr}
384 \fontseries{m}
385 \fontshape{n}
386 \fontsize{10}{10}

```

The initial `fontenc` package load list. This will get overwritten in `fonttext` and is only provided in case an old `fonttext.cfg` does not define the command:

```

387 \def\fontenc@load@list{\@elt{T1,OT1}}

```

We now load the customizable parts of NFSS.

```

388 \InputIfFileExists{fonttext.cfg}
389     {\typeout{=====^^J%
390             ^^J%
391             Local config file fonttext.cfg used^^J%
392             ^^J%
393             =====}%
394     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
395     }
396     {\input{fonttext.ltx}}
397 \let\@addtofilelist\@gobble

```

Ditto for math although I don't think that we will get a lot of customisation :-)

```

398 \InputIfFileExists{fontmath.cfg}
399     {\typeout{=====^^J%
400             ^^J%
401             Local config file fontmath.cfg used^^J%
402             ^^J%
403             =====}%
404     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
405     }
406     {\input{fontmath.ltx}}
407 \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`.

```

408 \InputIfFileExists{preload.cfg}
409     {\typeout{=====^^J%
410             ^^J%
411             Local config file preload.cfg used^^J%
412             ^^J%
413             =====}%
414     \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
415     }
416     {\input{preload.ltx}}
417 \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 418 \let\@acci\' \let\@accii\' \let\@acciii\=
```

\cal Here were the two old *<alphabet identifiers>*.

```
\mit
419 </2kernel>
```

File u

fontdef.dtx

j-latexreleasej [2020/02/11 v3.0g LaTeX Kernel (j-latexreleasej font setup)]

41 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.

42 Customization

You are not allowed to change this source file! If you want to change the default encodings and/or the font shape groups preloaded you should create a copy of `fonttext.ltx` under the name `fonttext.cfg` and change this copy. If $\text{\LaTeX} 2_{\epsilon}$ finds a file of this name it will use it, otherwise it uses the standard file which is `fontdef.ltx`.

If you don't plan to use Computer Modern much or at all, it might (!) be a good idea to make your own `fonttext.cfg`. Look at the comments below (docstrip module 'text') to see what should go into such a file.

To change the math font setup use a copy of `fontmath.ltx` under the name `fontmath.cfg` and change this copy. However, dealing with this interface is even more a job for an expert than changing the text font setup — in short, we don't encourage either.

Warning: please note that we don't support customised \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.

43 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 <code>fonttext.ltx</code>
math	produce the file <code>fontmath.ltx</code>
cfgtext	produce a dummy <code>fonttext.cfg</code> file
cfgmath	produce a dummy <code>fontmath.cfg</code> file

A typical `docstrip` command file would then have entries like:

```
\generateFile{fonttext.ltx}{t}{\from{fontdef.dtx}{text}}
```

44 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>

```

45 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}

```

45.1 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 $\langle enc \rangle enc.def$, 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 {omsenc.def}

```

Documents containing a lot of accented characters should really be using T1 fonts. We therefore load this last so that T1 encoding specific commands are executed as fast as possible (encoding files are no longer reloaded in `fontenc`).

```

12 \input {ot1enc.def}
13 \input {t1enc.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}

```

The initial `fontenc` package load list if an 8-bit \TeX engine is used:

```

17 \def\@fontenc@load@list{\@elt{T1,OT1}}
18 \def\rmsubstdefault{cmr}
19 \def\sfsubstdefault{cmss}
20 \def\ttsubstdefault{cmtt}
21 \LoadFontDefinitionFile{TS1}{cmr}
22 \else

```

Unicode.

```
23 \input {tuenc.def}
24 \fontencoding{TU}
    The initial fontenc package load list if a Unicode engine is used:
25 \def\@fontenc@load@list{\@elt{TU}}
26 \DeclareFontSubstitution{TU}{lmr}{m}{n}
27 \LoadFontDefinitionFile{TU}{lmr}
28 \LoadFontDefinitionFile{TU}{lms}
29 \LoadFontDefinitionFile{TU}{lmt}
30 \def\rmsubstdefault{lmr}
31 \def\sfsbstdefault{lms}
32 \def\ttsbstdefault{lmt}
33 \LoadFontDefinitionFile{TS1}{lmr}
34 \DeclareFontSubstitution{TU}{lmr}{m}{n}
```

End of Unicode branch.

```
35 \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!

```
36 \DeclareFontEncodingDefaults{}{}
```

Then we define the default substitution for every encoding. This release of L^AT_εX 2_ε 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.

```
37 \DeclareFontSubstitution{T1}{cmr}{m}{n}
38 \DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration, L^AT_εX 2_ε 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. L^AT_εX 2_ε will then check that for every encoding the substitution defaults form a valid font shape group, which means that it will check if there is a `\DeclareFontShape` declaration for this combination. We will therefore load the corresponding `.fd` files now. If we don't do this they would be loaded at verification time (i.e. at `\begin{document}`) which would delay processing unnecessarily.

Warning: Please note that this means that you have to regenerate the format whenever you change any of these `.fd` files since L^AT_εX 2_ε 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.

```
39 \begingroup
40 \nfss@catcodes
41 \input {t1cmr.fd}
```

```
42 \input {ot1cmr.fd}
43 \endgroup
```

We also load some other font definition files which are normally needed in a document. This is only done for processing speed and you can comment the next two lines out to save some memory. If necessary these files are then loaded when your document is processed. (Loading .fd files is a less drastic step compared to preloading fonts because the number of fonts is limited 255 at (nearly) every T_EX installation, while the amount of main memory is not a limiting factor at most installations.)

```
44 \begingroup
45 \nfss@catcodes
46 \input {ot1cmss.fd}
47 \input {ot1cmtt.fd}
48 \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.

```
49 \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.

45.2 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      50 \ifx\Umathchar\@undefined
  \ttdefault      51 \newcommand\encodingdefault{OT1}
                  52 \newcommand\rmdefault{cmr}
                  53 \newcommand\sfdefault{cmss}
                  54 \newcommand\ttdefault{cmtt}
                  55 \else
                  56 \newcommand\encodingdefault{TU}
                  57 \newcommand\rmdefault{lmr}
                  58 \fontfamily{\rmdefault}
                  59 \newcommand\sfdefault{lmss}
                  60 \newcommand\ttdefault{lmtt}
                  61 \fi
                  62 </text>
                  63 (latexrelease)\IncludeInRelease{2017/01/01}%
                  64 (latexrelease)          {\encodingdefault}{TU encoding default}%
                  65 (latexrelease)\ifx\Umathchar\@undefined
```

```

66 \latexrelease\renewcommand\encodingdefault{OT1}
67 \latexrelease\fontencoding{\encodingdefault}
68 \latexrelease\renewcommand\rmdefault{cmr}
69 \latexrelease\fontfamily{\rmdefault}
70 \latexrelease\renewcommand\sfddefault{cmss}
71 \latexrelease\renewcommand\ttdefault{cmtt}
72 \latexrelease\else
73 \latexrelease\renewcommand\encodingdefault{TU}
74 \latexrelease%done in everyjob\fontencoding{\encodingdefault}
75 \latexrelease\renewcommand\rmdefault{lmr}
76 \latexrelease\fontfamily{\rmdefault}
77 \latexrelease\renewcommand\sfddefault{lms}
78 \latexrelease\renewcommand\ttdefault{lmtt}
79 \latexrelease\fi
80 \latexrelease\EndIncludeInRelease
81 \latexrelease\IncludeInRelease{0000/00/00}%
82 \latexrelease          {\encodingdefault}{TU encoding default}%
83 \latexrelease\fontencoding{OT1}
84 \latexrelease\renewcommand\encodingdefault{OT1}
85 \latexrelease\fontencoding{\encodingdefault}
86 \latexrelease\renewcommand\rmdefault{cmr}
87 \latexrelease\fontfamily{\rmdefault}
88 \latexrelease\renewcommand\sfddefault{cmss}
89 \latexrelease\renewcommand\ttdefault{cmtt}
90 \latexrelease\EndIncludeInRelease
91 (*text)

```

`\bfdefault` Series changing commands are influenced by the following hooks.

```

\mddefault 92 \newcommand\bfdefault{b} % overwritten below
          93 \newcommand\mddefault{m}

```

`\itdefault` Shape changing commands use the following hooks.

```

\sldefault 94 \newcommand\itdefault{it}
\scdefault 95 \newcommand\sldefault{sl}
\updefault 96 \newcommand\scdefault{sc}
          97 \newcommand\updefault{up} % overwritten below

```

```

98 </text>
99 (*text | latexrelease)
100 \latexrelease\IncludeInRelease{2020/02/02}%
101 \latexrelease          {\updefault}{font defaults change}%
102 \renewcommand\updefault{up}
103 \renewcommand\bfdefault{b}
104 </text | latexrelease>
105 \latexrelease\EndIncludeInRelease
106 \latexrelease\IncludeInRelease{0000/00/00}%
107 \latexrelease          {\updefault}{font defaults change}%
108 \latexrelease
109 \latexrelease\renewcommand\updefault{n}
110 \latexrelease\renewcommand\bfdefault{bx}
111 \latexrelease\EndIncludeInRelease
112 (*text)

```

`\familydefault` Finally we have the hooks that describe the behaviour of the `\normalfont` command. To stay portable, the definition of `\encodingdefault` should *not* be
`\seriesdefault`
`\shapedefault`

changed and should match the setting above for `\fontencoding`. All other values can be set according to your taste.

```
113 \newcommand\familydefault{\rmdefault}
114 \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.

```
115 % \changes{v3.0e}{2019/12/17}{Set \cs{shapedefault} explicitly to ‘n’}
116 \newcommand\shapedefault{n}
```

This finishes the low-level setup in `fonttext.ltx`.

```
117 </text>
```

46 The fontmath.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
118 (*math)
119 \typeout{=== Don't modify this file, use a .cfg file instead ===^^J}
```

46.1 The font encodings used

```
120 \DeclareFontEncoding{OML}{}{}
121 \DeclareFontEncoding{OMS}{}{}
122 \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.

```
123 \DeclareFontEncoding{U}{}{\noaccents@}
```

The encodings for math are next:

```
124 \DeclareFontSubstitution{OML}{cmm}{m}{it}
125 \DeclareFontSubstitution{OMS}{cmsy}{m}{n}
126 \DeclareFontSubstitution{OMX}{cmex}{m}{n}
127 \DeclareFontSubstitution{U}{cmr}{m}{n}
```

```
128 \begingroup
129 \nfss@catcodes
130 \input {omlcmm.fd}
131 \input {omscmsy.fd}
132 \input {omxcmex.fd}
133 \input {ucmr.fd}
134 \endgroup
```

46.1.1 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.

```

135 \DeclareSymbolFont{operators}    {OT1}{cmr} {m}{n}
136 \DeclareSymbolFont{letters}     {OML}{cmm} {m}{it}
137 \DeclareSymbolFont{symbols}     {OMS}{cmsy}{m}{n}
138 \DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}

139 \SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}
140 \SetSymbolFont{letters}  {bold}{OML}{cmm} {b}{it}
141 \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.

```

142 \DeclareSymbolFontAlphabet{\mathrm}    {operators}
143 \DeclareSymbolFontAlphabet{\mathnormal}{letters}
144 \DeclareSymbolFontAlphabet{\mathcal}   {symbols}
145 \DeclareMathAlphabet      {\mathbf}    {OT1}{cmr}{bx}{n}
146 \DeclareMathAlphabet      {\mathsf}    {OT1}{cmss}{m}{n}
147 \DeclareMathAlphabet      {\mathit}    {OT1}{cmr}{m}{it}
148 \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.

```

149 \SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
150 \SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}

```

46.2 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).

```

151 \DeclareMathSizes{5}{5}{5}{5}
152 \DeclareMathSizes{6}{6}{5}{5}
153 \DeclareMathSizes{7}{7}{5}{5}
154 \DeclareMathSizes{8}{8}{6}{5}
155 \DeclareMathSizes{9}{9}{6}{5}
156 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
157 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
158 \DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
159 \DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
160 \DeclareMathSizes{\@xvipt}{\@xvipt}{\@xiipt}{\@xpt}
161 \DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
162 \DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xvipt}

```

46.3 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 Ini \TeX . However, we repeat them here to have a complete setup which can be exchanged with another if desired.

46.3.1 The letters

```

163 \DeclareMathSymbol{a}{\mathalpha}{letters}{'a}
164 \DeclareMathSymbol{b}{\mathalpha}{letters}{'b}
165 \DeclareMathSymbol{c}{\mathalpha}{letters}{'c}
166 \DeclareMathSymbol{d}{\mathalpha}{letters}{'d}
167 \DeclareMathSymbol{e}{\mathalpha}{letters}{'e}
168 \DeclareMathSymbol{f}{\mathalpha}{letters}{'f}
169 \DeclareMathSymbol{g}{\mathalpha}{letters}{'g}
170 \DeclareMathSymbol{h}{\mathalpha}{letters}{'h}
171 \DeclareMathSymbol{i}{\mathalpha}{letters}{'i}
172 \DeclareMathSymbol{j}{\mathalpha}{letters}{'j}
173 \DeclareMathSymbol{k}{\mathalpha}{letters}{'k}
174 \DeclareMathSymbol{l}{\mathalpha}{letters}{'l}
175 \DeclareMathSymbol{m}{\mathalpha}{letters}{'m}
176 \DeclareMathSymbol{n}{\mathalpha}{letters}{'n}
177 \DeclareMathSymbol{o}{\mathalpha}{letters}{'o}
178 \DeclareMathSymbol{p}{\mathalpha}{letters}{'p}
179 \DeclareMathSymbol{q}{\mathalpha}{letters}{'q}
180 \DeclareMathSymbol{r}{\mathalpha}{letters}{'r}
181 \DeclareMathSymbol{s}{\mathalpha}{letters}{'s}
182 \DeclareMathSymbol{t}{\mathalpha}{letters}{'t}
183 \DeclareMathSymbol{u}{\mathalpha}{letters}{'u}
184 \DeclareMathSymbol{v}{\mathalpha}{letters}{'v}
185 \DeclareMathSymbol{w}{\mathalpha}{letters}{'w}
186 \DeclareMathSymbol{x}{\mathalpha}{letters}{'x}
187 \DeclareMathSymbol{y}{\mathalpha}{letters}{'y}
188 \DeclareMathSymbol{z}{\mathalpha}{letters}{'z}

189 \DeclareMathSymbol{A}{\mathalpha}{letters}{'A}
190 \DeclareMathSymbol{B}{\mathalpha}{letters}{'B}
191 \DeclareMathSymbol{C}{\mathalpha}{letters}{'C}
192 \DeclareMathSymbol{D}{\mathalpha}{letters}{'D}
193 \DeclareMathSymbol{E}{\mathalpha}{letters}{'E}
194 \DeclareMathSymbol{F}{\mathalpha}{letters}{'F}
195 \DeclareMathSymbol{G}{\mathalpha}{letters}{'G}
196 \DeclareMathSymbol{H}{\mathalpha}{letters}{'H}
197 \DeclareMathSymbol{I}{\mathalpha}{letters}{'I}
198 \DeclareMathSymbol{J}{\mathalpha}{letters}{'J}
199 \DeclareMathSymbol{K}{\mathalpha}{letters}{'K}
200 \DeclareMathSymbol{L}{\mathalpha}{letters}{'L}
201 \DeclareMathSymbol{M}{\mathalpha}{letters}{'M}
202 \DeclareMathSymbol{N}{\mathalpha}{letters}{'N}
203 \DeclareMathSymbol{O}{\mathalpha}{letters}{'O}
204 \DeclareMathSymbol{P}{\mathalpha}{letters}{'P}
205 \DeclareMathSymbol{Q}{\mathalpha}{letters}{'Q}
206 \DeclareMathSymbol{R}{\mathalpha}{letters}{'R}
207 \DeclareMathSymbol{S}{\mathalpha}{letters}{'S}
208 \DeclareMathSymbol{T}{\mathalpha}{letters}{'T}
209 \DeclareMathSymbol{U}{\mathalpha}{letters}{'U}
210 \DeclareMathSymbol{V}{\mathalpha}{letters}{'V}
211 \DeclareMathSymbol{W}{\mathalpha}{letters}{'W}

```

```

212 \DeclareMathSymbol{X}{\mathalpha}{letters}{‘X}
213 \DeclareMathSymbol{Y}{\mathalpha}{letters}{‘Y}
214 \DeclareMathSymbol{Z}{\mathalpha}{letters}{‘Z}

```

46.3.2 The digits

```

215 \DeclareMathSymbol{0}{\mathalpha}{operators}{‘0}
216 \DeclareMathSymbol{1}{\mathalpha}{operators}{‘1}
217 \DeclareMathSymbol{2}{\mathalpha}{operators}{‘2}
218 \DeclareMathSymbol{3}{\mathalpha}{operators}{‘3}
219 \DeclareMathSymbol{4}{\mathalpha}{operators}{‘4}
220 \DeclareMathSymbol{5}{\mathalpha}{operators}{‘5}
221 \DeclareMathSymbol{6}{\mathalpha}{operators}{‘6}
222 \DeclareMathSymbol{7}{\mathalpha}{operators}{‘7}
223 \DeclareMathSymbol{8}{\mathalpha}{operators}{‘8}
224 \DeclareMathSymbol{9}{\mathalpha}{operators}{‘9}

```

46.3.3 Punctuation, brace, etc. keys

```

225 \DeclareMathSymbol{!}{\mathclose}{operators}{"21}
226 \DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
227 \DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
228 \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
229 \DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
230 \DeclareMathSymbol{.}{\mathord}{letters}{"3A}
231 \DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
232 \DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
233 \DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
234 \DeclareMathSymbol{?}{\mathclose}{operators}{"3F}

```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```

235 %\DeclareMathSymbol{(}{\mathopen}{operators}{"28}
236 %\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
237 %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
238 %\DeclareMathSymbol{[}{\mathopen}{operators}{"5B}
239 %\DeclareMathSymbol{]}{\mathclose}{operators}{"5D}
240 %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
241 %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
242 %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}

```

Should all of the following being activated by default? Probably not.

```

243 %\DeclareMathSymbol{\}{\mathopen}{symbols}{"66}
244 %\DeclareMathSymbol{'}{\mathclose}{symbols}{"67}
245 %\DeclareMathSymbol{\'}{\mathord}{symbols}{"6E} % \backslash
246 \mathcode\ = "8000 % \space
247 \mathcode\ ' = "8000 % ^\prime
248 \mathcode\ _ = "8000 % \_

```

46.3.4 Delimitercodes for characters

[to be completed]

Finally, IniT_EX sets all `\delcode` values to -1, except `\delcode‘.=0`

```

249 \DeclareMathDelimiter{(}{\mathopen} {operators}{"28}{largesymbols}{"00}
250 \DeclareMathDelimiter{)}{\mathclose}{operators}{"29}{largesymbols}{"01}
251 \DeclareMathDelimiter{[}{\mathopen} {operators}{"5B}{largesymbols}{"02}
252 \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.

```
253 \DeclareMathDelimiter{<}{\mathopen}{symbols}{"68}{largesymbols}{"0A}
254 \DeclareMathDelimiter{>}{\mathclose}{symbols}{"69}{largesymbols}{"0B}
255 \DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
256 \DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
```

And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```
257 \DeclareMathDelimiter{/}{\mathord}{operators}{"2F}{largesymbols}{"0E}
258 \DeclareMathSymbol{/}{\mathord}{letters}{"3D}

259 \DeclareMathDelimiter{|}{\mathord}{symbols}{"6A}{largesymbols}{"0C}
260 \expandafter\DeclareMathDelimiter\@backslashchar
261 \mathord{symbols}{"6E}{largesymbols}{"0F}
```

N.B. { and } should NOT get delcodes; otherwise parameter grouping fails!

46.4 Symbols accessed via control sequences

46.4.1 Greek letters

```
262 \DeclareMathSymbol{\alpha}{\mathord}{letters}{"0B}
263 \DeclareMathSymbol{\beta}{\mathord}{letters}{"0C}
264 \DeclareMathSymbol{\gamma}{\mathord}{letters}{"0D}
265 \DeclareMathSymbol{\delta}{\mathord}{letters}{"0E}
266 \DeclareMathSymbol{\epsilon}{\mathord}{letters}{"0F}
267 \DeclareMathSymbol{\zeta}{\mathord}{letters}{"10}
268 \DeclareMathSymbol{\eta}{\mathord}{letters}{"11}
269 \DeclareMathSymbol{\theta}{\mathord}{letters}{"12}
270 \DeclareMathSymbol{\iota}{\mathord}{letters}{"13}
271 \DeclareMathSymbol{\kappa}{\mathord}{letters}{"14}
272 \DeclareMathSymbol{\lambda}{\mathord}{letters}{"15}
273 \DeclareMathSymbol{\mu}{\mathord}{letters}{"16}
274 \DeclareMathSymbol{\nu}{\mathord}{letters}{"17}
275 \DeclareMathSymbol{\xi}{\mathord}{letters}{"18}
276 \DeclareMathSymbol{\pi}{\mathord}{letters}{"19}
277 \DeclareMathSymbol{\rho}{\mathord}{letters}{"1A}
278 \DeclareMathSymbol{\sigma}{\mathord}{letters}{"1B}
279 \DeclareMathSymbol{\tau}{\mathord}{letters}{"1C}
280 \DeclareMathSymbol{\upsilon}{\mathord}{letters}{"1D}
281 \DeclareMathSymbol{\phi}{\mathord}{letters}{"1E}
282 \DeclareMathSymbol{\chi}{\mathord}{letters}{"1F}
283 \DeclareMathSymbol{\psi}{\mathord}{letters}{"20}
284 \DeclareMathSymbol{\omega}{\mathord}{letters}{"21}
285 \DeclareMathSymbol{\varepsilon}{\mathord}{letters}{"22}
286 \DeclareMathSymbol{\vartheta}{\mathord}{letters}{"23}
287 \DeclareMathSymbol{\varpi}{\mathord}{letters}{"24}
288 \DeclareMathSymbol{\varrho}{\mathord}{letters}{"25}
289 \DeclareMathSymbol{\varsigma}{\mathord}{letters}{"26}
290 \DeclareMathSymbol{\varphi}{\mathord}{letters}{"27}
291 \DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{"00}
```

```

292 \DeclareMathSymbol{\Delta}{\mathalpha}{operators}{"01}
293 \DeclareMathSymbol{\Theta}{\mathalpha}{operators}{"02}
294 \DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{"03}
295 \DeclareMathSymbol{\Xi}{\mathalpha}{operators}{"04}
296 \DeclareMathSymbol{\Pi}{\mathalpha}{operators}{"05}
297 \DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{"06}
298 \DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{"07}
299 \DeclareMathSymbol{\Phi}{\mathalpha}{operators}{"08}
300 \DeclareMathSymbol{\Psi}{\mathalpha}{operators}{"09}
301 \DeclareMathSymbol{\Omega}{\mathalpha}{operators}{"0A}

```

46.4.2 Ordinary symbols

```

302 \DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
303 \DeclareMathSymbol{\imath}{\mathord}{letters}{"7B}
304 \DeclareMathSymbol{\jmath}{\mathord}{letters}{"7C}
305 \DeclareMathSymbol{\ell}{\mathord}{letters}{"60}
306 \DeclareMathSymbol{\wp}{\mathord}{letters}{"7D}
307 \DeclareMathSymbol{\Re}{\mathord}{symbols}{"3C}
308 \DeclareMathSymbol{\Im}{\mathord}{symbols}{"3D}
309 \DeclareMathSymbol{\partial}{\mathord}{letters}{"40}
310 \DeclareMathSymbol{\infty}{\mathord}{symbols}{"31}
311 \DeclareMathSymbol{\prime}{\mathord}{symbols}{"30}
312 \DeclareMathSymbol{\emptyset}{\mathord}{symbols}{"3B}
313 \DeclareMathSymbol{\nabla}{\mathord}{symbols}{"72}
314 \DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
315 \DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
316 \DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
317 \DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
318 \DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
319 \DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}

```

Alias:

```

320 % \let\not=\neg
321 \DeclareMathSymbol{\lnot}{\mathord}{symbols}{"3A}
322 \DeclareMathSymbol{\flat}{\mathord}{letters}{"5B}
323 \DeclareMathSymbol{\natural}{\mathord}{letters}{"5C}
324 \DeclareMathSymbol{\sharp}{\mathord}{letters}{"5D}
325 \DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{"7C}
326 \DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{"7D}
327 \DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{"7E}
328 \DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{"7F}

329 \DeclareRobustCommand\hbar{{\mathchar'26\mkern-9mu}}
330 \DeclareRobustCommand\surd{{\mathchar"1270}}
331 \DeclareRobustCommand\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crrc
332 \not\mathrel{\mkern14mu}\crrc
333 \noalign{\nointerlineskip}
334 \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crrc}}}}

```

46.4.3 Large Operators

```

335 \DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{"60}
336 \DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{"57}
337 \DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{"56}
338 \DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{"55}
339 \DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{"54}

```

```

340 \DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{"53}
341 \DeclareMathSymbol{\intop}{\mathop}{largesymbols}{"52}
342   \DeclareRobustCommand\int{\intop\nolimits}
343 \DeclareMathSymbol{\prod}{\mathop}{largesymbols}{"51}
344 \DeclareMathSymbol{\sum}{\mathop}{largesymbols}{"50}
345 \DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{"4E}
346 \DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{"4C}
347 \DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{"4A}
348 \DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{"48}
349   \DeclareRobustCommand\oint{\ointop\nolimits}
350 \DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{"46}
351 \DeclareMathSymbol{\smallint}{\mathop}{symbols}{"73}

```

46.4.4 Binary symbols

```

352 \DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{"2F}
353 \DeclareMathSymbol{\triangleright}{\mathbin}{letters}{"2E}
354 \DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{"34}
355 \DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{"35}

```

Alias:

```

356 %   \let \varbigtriangledown \bigtriangledown
357 %   \let \varbigtriangleup \bigtriangleup
358 \DeclareMathSymbol{\varbigtriangleup}{\mathbin}{symbols}{"34}
359 \DeclareMathSymbol{\varbigtriangledown}{\mathbin}{symbols}{"35}

```

These last two synonyms are needed because the stmaryd package redefines them as Operators.

```

360 \DeclareMathSymbol{\wedge}{\mathbin}{symbols}{"5E}
361 \DeclareMathSymbol{\vee}{\mathbin}{symbols}{"5F}

```

Alias:

```

362 %   \let \land=\wedge
363 %   \let \lor=\vee
364 \DeclareMathSymbol{\land}{\mathbin}{symbols}{"5E}
365 \DeclareMathSymbol{\lor}{\mathbin}{symbols}{"5F}
366 \DeclareMathSymbol{\cap}{\mathbin}{symbols}{"5C}
367 \DeclareMathSymbol{\cup}{\mathbin}{symbols}{"5B}
368 \DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{"7A}
369 \DeclareMathSymbol{\dagger}{\mathbin}{symbols}{"79}
370 \DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{"75}
371 \DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{"74}
372 \DeclareMathSymbol{\uplus}{\mathbin}{symbols}{"5D}
373 \DeclareMathSymbol{\amalg}{\mathbin}{symbols}{"71}
374 \DeclareMathSymbol{\diamond}{\mathbin}{symbols}{"05}
375 \DeclareMathSymbol{\bullet}{\mathbin}{symbols}{"0F}
376 \DeclareMathSymbol{\wr}{\mathbin}{symbols}{"6F}
377 \DeclareMathSymbol{\div}{\mathbin}{symbols}{"04}
378 \DeclareMathSymbol{\odot}{\mathbin}{symbols}{"0C}
379 \DeclareMathSymbol{\oslash}{\mathbin}{symbols}{"0B}
380 \DeclareMathSymbol{\otimes}{\mathbin}{symbols}{"0A}
381 \DeclareMathSymbol{\ominus}{\mathbin}{symbols}{"09}
382 \DeclareMathSymbol{\oplus}{\mathbin}{symbols}{"08}
383 \DeclareMathSymbol{\mp}{\mathbin}{symbols}{"07}
384 \DeclareMathSymbol{\pm}{\mathbin}{symbols}{"06}
385 \DeclareMathSymbol{\circ}{\mathbin}{symbols}{"0E}

```

```

386 \DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{"0D}
387 \DeclareMathSymbol{\setminus}{\mathbin}{symbols}{"6E}
388 \DeclareMathSymbol{\cdot}{\mathbin}{symbols}{"01}
389 \DeclareMathSymbol{\ast}{\mathbin}{symbols}{"03}
390 \DeclareMathSymbol{\times}{\mathbin}{symbols}{"02}
391 \DeclareMathSymbol{\star}{\mathbin}{letters}{"3F}

```

46.4.5 Relations

```

392 \DeclareMathSymbol{\propto}{\mathrel}{symbols}{"2F}
393 \DeclareMathSymbol{\sqsubset}{\mathrel}{symbols}{"76}
394 \DeclareMathSymbol{\sqsupset}{\mathrel}{symbols}{"77}
395 \DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
396 \DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
397 \DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
398 \DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
399 \DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
400 \DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
401 \DeclareMathSymbol{\nrightarrow}{\mathrel}{symbols}{"2D}
402 \DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
403 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
404 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
405 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
406 \DeclareRobustCommand\neq{\not=}

```

As `\neq` is robust we should not use `\let` to define `\ne` as then it would change if `\neq` changes.

```
407 \DeclareRobustCommand\ne{\not=}
```

It would be ok to use `\let` for those declared by `\DeclareMathSymbol` but for a cleaner interface we avoid it always (just in case the internals change).

```

408 \DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
409 \DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}

```

Alias:

```

410 % \let\le=\leq
411 % \let\ge=\geq
412 \DeclareMathSymbol{\le}{\mathrel}{symbols}{"14}
413 \DeclareMathSymbol{\ge}{\mathrel}{symbols}{"15}
414 \DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
415 \DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
416 \DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
417 \DeclareMathSymbol{\succeed}{\mathrel}{symbols}{"17}
418 \DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
419 \DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
420 \DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
421 \DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
422 \DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
423 \DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
424 \DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}

```

Alias:

```

425 % \let\owns=\ni
426 \DeclareMathSymbol{\owns}{\mathrel}{symbols}{"33}
427 \DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
428 \DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
429 \DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}

```

```

430 \DeclareMathSymbol{\leftrightharrow}{\mathrel}{symbols}{"24}
431 \DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
432 \DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}

Alias:
433 % \let\gets=\leftarrow
434 % \let\to=\rightarrow
435 \DeclareMathSymbol{\gets}{\mathrel}{symbols}{"20}
436 \DeclareMathSymbol{\to}{\mathrel}{symbols}{"21}
437 \DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
438 \DeclareRobustCommand\mapsto{\mapstochar\rightarrow}
439 \DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
440 \DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
441 \DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
442 \DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
443 \DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
444 \DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}
445 \DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
446 \DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
447 \DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
448 \DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
449 \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.

```

450 \DeclareRobustCommand
451 \cong{\mathrel{\mathpalette@vereq\sim}} % congruence sign
452 \def@vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
453 \ialign{${\m@th#1\hfil##\hfil$\crcr#2\crcr=}\crcr}}
454 \DeclareRobustCommand
455 \notin{\mathrel{\m@th\mathpalette@cncel\in}}
456 \def@cncel#1#2{\m@th\oalign{${\hfil#1\mkern1mu}\hfil$\crcr$#1#2$}}
457 \DeclareRobustCommand
458 \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
459 \def\rlh@#1{\vcenter{\m@th\hbox{\oalign{\raise2pt
460 \hbox{${#1}\rightharpoonup$}\crcr
461 $#1\leftharpoondown$}}}}
462 \DeclareRobustCommand
463 \doteq{\buildrel\textstyle.\over=}

```

46.4.6 Arrows

```

464 \DeclareRobustCommand
465 \joinrel{\mathrel{\mkern-3mu}}
466 \DeclareRobustCommand
467 \relbar{\mathrel{\smash-}} % \smash, because -
468 % 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).

```

469 \DeclareRobustCommand
470   \Relbar{\mathrel{=}}
471 \DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
472   \DeclareRobustCommand\hookrightarrow{\lhook\joinrel\rightarrow}
473 \DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
474   \DeclareRobustCommand\hookleftarrow{\leftarrow\joinrel\rhook}
475 \DeclareRobustCommand
476   \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
477 \DeclareRobustCommand
478   \models{\mathrel{|}\joinrel\Relbar}
479 \DeclareRobustCommand
480   \Longrightarrow{\Relbar\joinrel\Rightarrow}

```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make then robust.

```

481 \DeclareRobustCommand\longrightarrow
482   {\relbar\joinrel\rightarrow}
483 \DeclareRobustCommand\longleftarrow
484   {\leftarrow\joinrel\relbar}
485 \DeclareRobustCommand
486   \Longleftarrow{\Leftarrow\joinrel\Relbar}
487 \DeclareRobustCommand
488   \longmapsto{\mapstochar\longrightarrow}
489 \DeclareRobustCommand
490   \longleftrightarrow{\leftarrow\joinrel\rightarrow}
491 \DeclareRobustCommand
492   \Longleftrightarrow{\Leftarrow\joinrel\Rightarrow}
493 \DeclareRobustCommand
494   \iff{\;\Longleftrightarrow\;}

```

46.4.7 Punctuation symbols

```

495 \DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
496 \DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
497 \DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}

```

This is commented out, since `\ldots` is now defined in `ltoutenc.dtx`.

```

498 %\def\ldots{\mathinner{\ldotp\ldotp\ldotp}}
499 %\DeclareRobustCommand\ldots
500 %   {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots\,$}\fi}
501 \DeclareRobustCommand
502   \cdots{\mathinner{\cdotp\cdotp\cdotp}}
503 \DeclareRobustCommand
504   \vdots{\vbox{\baselineskip4\p@ \lineskiplimit\z@
505     \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
506 \DeclareRobustCommand
507   \ddots{\mathinner{\mkern1mu\raise7\p@
508     \vbox{\kern7\p@\hbox{.}}\mkern2mu
509     \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}

```

46.4.8 Math accents

```

510 \DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
511 \DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}

```

```

512 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
513 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
514 \DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
515 \DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
516 \DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
517 \DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
518 \DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
519 \DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
520 \DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
521 \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:

```

522 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{"17}

```

46.4.9 Radicals

```

523 \DeclareMathRadical{\sqrtsign}{symbols}{"70}{largesymbols}{"70}

```

46.4.10 Over and under something, etc

```

524 \DeclareRobustCommand\overrightarrow[1]{\vbox{\m@th\ialign{##\crrc
525     \rightarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}
526     $\hfil\displaystyle{#1}\hfil$\crrc}}}
527 \DeclareRobustCommand\overleftarrow[1]{\vbox{\m@th\ialign{##\crrc
528     \leftarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}%
529     $\hfil\displaystyle{#1}\hfil$\crrc}}}
530 \DeclareRobustCommand\overbrace[1]
531     {\mathop{\vbox{\m@th\ialign{##\crrc\noalign{\kern3\p@}%
532     \downbracefill\crrc\noalign{\kern3\p@\nointerlineskip}%
533     $\hfil\displaystyle{#1}\hfil$\crrc}}}\limits}
534 \DeclareRobustCommand\underbrace[1]{\mathop{\vtop{\m@th\ialign{##\crrc
535     $\hfil\displaystyle{#1}\hfil$\crrc
536     \noalign{\kern3\p@\nointerlineskip}%
537     \upbracefill\crrc\noalign{\kern3\p@}}}\limits}

```

(quite a waste of tokens, IMHO — Frank)

```

538 \DeclareRobustCommand\skew[3]
539     {\{\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
540     #2{\mkern-\muskip\z@#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
541 \DeclareRobustCommand\rightarrowfill{${\m@th\smash-\mkern-7mu%
542 \cleaders\hbox{${\mkern-2mu\smash-\mkern-2mu}$}\hfill
543 \mkern-7mu\mathord\rightarrow$}
544 \DeclareRobustCommand\leftarrowfill{${\m@th\mathord\leftarrow\mkern-7mu%
545 \cleaders\hbox{${\mkern-2mu\smash-\mkern-2mu}$}\hfill
546 \mkern-7mu\smash-$}
547 \DeclareMathSymbol{\braceld}{\mathord}{largesymbols}{"7A}
548 \DeclareMathSymbol{\bracerd}{\mathord}{largesymbols}{"7B}
549 \DeclareMathSymbol{\bracelu}{\mathord}{largesymbols}{"7C}
550 \DeclareMathSymbol{\braceru}{\mathord}{largesymbols}{"7D}
551 \DeclareRobustCommand\downbracefill{${\m@th \setbox\z@\hbox{${\braceld$}%
552 \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru
553 \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd$}
554 \DeclareRobustCommand\upbracefill{${\m@th \setbox\z@\hbox{${\braceld$}%
555 \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
556 \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}

```

46.4.11 Delimiters

```

557 \DeclareMathDelimiter{\lmoustache} % top from (, bottom from )
558   {\mathopen}{largesymbols}{"7A}{largesymbols}"40}
559 \DeclareMathDelimiter{\rmoustache} % top from ), bottom from (
560   {\mathclose}{largesymbols}"7B}{largesymbols}"41}
561 \DeclareMathDelimiter{\arrowvert} % arrow without arrowheads
562   {\mathord}{symbols}"6A}{largesymbols}"3C}
563 \DeclareMathDelimiter{\Arrowvert} % double arrow without arrowheads
564   {\mathord}{symbols}"6B}{largesymbols}"3D}
565 \DeclareMathDelimiter{\Vert}
566   {\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
567 %\let\|= \Vert
568 \DeclareMathDelimiter{\|}
569   {\mathord}{symbols}"6B}{largesymbols}"0D}
570 \DeclareMathDelimiter{\vert}
571   {\mathord}{symbols}"6A}{largesymbols}"0C}
572 \DeclareMathDelimiter{\uparrow}
573   {\mathrel}{symbols}"22}{largesymbols}"78}
574 \DeclareMathDelimiter{\downarrow}
575   {\mathrel}{symbols}"23}{largesymbols}"79}
576 \DeclareMathDelimiter{\updownarrow}
577   {\mathrel}{symbols}"6C}{largesymbols}"3F}
578 \DeclareMathDelimiter{\Uparrow}
579   {\mathrel}{symbols}"2A}{largesymbols}"7E}
580 \DeclareMathDelimiter{\Downarrow}
581   {\mathrel}{symbols}"2B}{largesymbols}"7F}
582 \DeclareMathDelimiter{\Updownarrow}
583   {\mathrel}{symbols}"6D}{largesymbols}"77}
584 \DeclareMathDelimiter{\backslash} % for double coset G\backslash H
585   {\mathord}{symbols}"6E}{largesymbols}"0F}
586 \DeclareMathDelimiter{\rangle}
587   {\mathclose}{symbols}"69}{largesymbols}"0B}
588 \DeclareMathDelimiter{\langle}
589   {\mathopen}{symbols}"68}{largesymbols}"0A}
590 \DeclareMathDelimiter{\rbrace}
591   {\mathclose}{symbols}"67}{largesymbols}"09}
592 \DeclareMathDelimiter{\lbrace}
593   {\mathopen}{symbols}"66}{largesymbols}"08}
594 \DeclareMathDelimiter{\rceil}
595   {\mathclose}{symbols}"65}{largesymbols}"07}
596 \DeclareMathDelimiter{\lceil}
597   {\mathopen}{symbols}"64}{largesymbols}"06}
598 \DeclareMathDelimiter{\rfloor}
599   {\mathclose}{symbols}"63}{largesymbols}"05}
600 \DeclareMathDelimiter{\lfloor}
601   {\mathopen}{symbols}"62}{largesymbols}"04}

```

`\lgroup` There are three plain T_EX 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.)

```
602 \DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
603     {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
604 \DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
605     {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}
606 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
607     {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}
```

46.5 Math versions of text commands

The `\mathunderscore` here is really a text definition, so it has been put back into `loutenc.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 608 \DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
\mathdollar 609 \DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
\mathsterling 610 \DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
\mathunderscore 611 \DeclareRobustCommand\mathsterling{\mathit{\mathchar"7024}}
612 \DeclareRobustCommand\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}

\mathellipsis This is plain TEX's \ldots.
613 \DeclareRobustCommand\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%
```

46.6 Other special functions and parameters

46.6.1 Biggggg

```
614 </math>
615 <{*math | latexrelease}
616 <latexrelease>\IncludeInRelease{2018/12/01}%
617 <latexrelease>           {\Big}{Start LR-mode}%
618 \DeclareRobustCommand\big[1]{\leavevmode@ifvmode
619     {\hbox{\$left#1\vbox to8.5p@{\}right.\n@space$}}}
620 \DeclareRobustCommand\Big[1]{\leavevmode@ifvmode
621     {\hbox{\$left#1\vbox to11.5p@{\}right.\n@space$}}}
622 \DeclareRobustCommand\bigg[1]{\leavevmode@ifvmode
623     {\hbox{\$left#1\vbox to14.5p@{\}right.\n@space$}}}
624 \DeclareRobustCommand\Bigg[1]{\leavevmode@ifvmode
625     {\hbox{\$left#1\vbox to17.5p@{\}right.\n@space$}}}
626 </math | latexrelease>
627 <latexrelease>\EndIncludeInRelease
628 <latexrelease>\IncludeInRelease{0000/00/00}%
629 <latexrelease>           {\Big}{Start LR-mode}%
630 <latexrelease>\def\big#1{{\hbox{\$left#1\vbox to8.5p@{\}right.\n@space$}}}
631 <latexrelease>\def\Big#1{{\hbox{\$left#1\vbox to11.5p@{\}right.\n@space$}}}
632 <latexrelease>\def\bigg#1{{\hbox{\$left#1\vbox to14.5p@{\}right.\n@space$}}}
633 <latexrelease>\def\Bigg#1{{\hbox{\$left#1\vbox to17.5p@{\}right.\n@space$}}}
634 <latexrelease>\EndIncludeInRelease
635 <{*math}>
```

```
636 \def\n@space{\nulldelimiterspace\z@ \m@th}
```

46.6.2 The log-like functions

`\operator@font` The `\operator@font` determines the symbol font used for log-like functions.

```
637 \def\operator@font{\mathgroup\symoperators}
```

46.6.3 Parameters

```
638 \thinmuskip=3mu
```

```
639 \medmuskip=4mu plus 2mu minus 4mu
```

```
640 \thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in `fontmath.ltx`.

```
641 </math>
```

47 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.

```
642 (*cfgtext | cfgmath | cfgprel)
```

```
643 %%
```

```
644 %%
```

```
645 %%
```

```
646 %% Load the standard setup:
```

```
647 %%
```

```
648 (+cfgtext)\input{fonttext.ltx}
```

```
649 (+cfgmath)\input{fontmath.ltx}
```

```
650 (+cfgprel)\input{preload.ltx}
```

```
651 %%
```

```
652 %% Small changes could go here; see documentation in cfgguide.tex for
```

```
653 %% allowed modifications.
```

```
654 %%
```

```
655 %% In particular it is not allowed to misuse this configuration file
```

```
656 %% to modify internal LaTeX commands!
```

```
657 %%
```

```
658 %% If you use this file as the basis for configuration please change
```

```
659 %% the \ProvidesFile lines to clearly identify your modification, e.g.,
```

```
660 %%
```

```
661 (+cfgtext)%% \ProvidesFile{fonttext.cfg}[2001/06/01
```

```
662 (+cfgmath)%% \ProvidesFile{fontmath.cfg}[2001/06/01
```

```
663 (+cfgprel)%% \ProvidesFile{preload.cfg}[2001/06/01
```

```
664 %% Customised local font setup]
```

```
665 %%
```

```
666 %%
```

```
667 </cfgtext | cfgmath | cfgprel)
```

File v

preload.dtx

48 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by $\text{\LaTeX} 2_{\epsilon}$). 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
dpreloa.xpt	preload of DC fonts for 10pt size
dpreloa.xip	preload of DC fonts for 11pt size
dpreloa.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.

49 Customization

You can customize the preloaded fonts in your $\text{\LaTeX} 2_{\epsilon}$ 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*` $\text{\LaTeX} 2_{\epsilon}$ 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 $\text{\LaTeX} 2.09$.

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.

50 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.

51 A driver for this document

The next bit of code contains the documentation driver file for \TeX , i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 %\OnlyDescription % comment out for implementation details
4 \begin{document}
5   \DocInput{preload.dtx}
6 \end{document}
7 </driver>
```

52 The code

We begin by loading the math extension font (`cmex10`) and the \LaTeX line and circle fonts. It is necessary to do this explicitly since these are used by `lplain.tex` and `latex.tex`. Since the internal font name contains / characters and digits we construct the name via `\csname`. These are the only fonts (!) that must be loaded in this file.

All `\DeclarePreloadSizes` can be removed or others can be added, they only influence the processing speed.

```
8 \expandafter\font\csname OMX/cmex/m/n/10\endcsname=cmex10\relax
9 \font\tenln =line10 \font\tenlnw =linew10\relax
10 \font\tenclrc=lcircle10 \font\tenclrcw=lcirclew10\relax
```

The above fonts should not be touched but anything below this point here in the preload suggestions can be modified without any problems.

```
11 (-tex)%*****
```

```

12 (-tex)% Start any modification below this point **
13 (-tex)%*****
14 (-tex)
15 %%
16 %% Computer Modern Roman:
17 %%-----
18 (*ori)
19 \DeclarePreloadSizes{OT1}{cmr}{m}{n}
20     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
21 \DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
22 \DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
23 \DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
24 (/ori)
25 (+xpt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
26 (+xpt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
27 (+xipt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
28 (+xipt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
29 (+xiipt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
30 (+xiipt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
31 %%
32 %% Computer Modern Sans:
33 %%-----
34 (+ori) \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
35 %%
36 %% Computer Modern Typewriter:
37 %%-----
38 (+ori) \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
39 %%
40 %% Computer Modern Math:
41 %%-----
42 (*ori)
43 \DeclarePreloadSizes{OML}{cmm}{m}{it}
44     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
45 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}
46     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
47 (/ori)

```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```

48 (*xpt)
49 \DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
50 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
51 (/xpt)
52 (*xipt)
53 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
54 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
55 (/xipt)
56 (*xiipt)
57 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
58 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
59 (/xiipt)
60 %%
61 %% LaTeX symbol fonts:
62 %%-----

```

```
63 <*ori>
64 \DeclarePreloadSizes{U}{\lasy}{m}{n}
65     {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
66 </ori>
67 </preload>
```

File w

lftntcmd.dtx

Abstract

The commands defined in this file `lftntcmd` are part of the kernel code for $\text{\LaTeX} 2_{\epsilon}/\text{NFSS2}$.

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.

53 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 $\text{T}_{\text{E}}\text{X}$ system and for several reasons it is better to avoid them on the user level whenever possible. In $\text{\LaTeX}3$ 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`.⁹ The default definition for this list is produced by the following.

```
\newcommand \nocorrlist {,}
```

⁹Any 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 `\/`.

54 The implementation

`\DeclareTextFontCommand` This is the creator function for `\text..` commands. It gives a warning if `\foo` or `\fragfoo` is already defined.

In math mode it simply puts the font declaration and text into a box (possibly an automagically sized one).

Otherwise it first scans the text to see where `\nocorr` occurs within it. This sets the `\check@ic` commands to do what is necessary concerning the italic correction at both ends.

The algorithm for deciding whether to put in an italic correction is not very subtle: one is added whenever the newly current font is not itself positively sloped, unless the next token is a character in the ‘nocorr’ list. At the end of the text this is done after closing the group so as to check the ‘outer font’. Note that this will often result in adding an italic correction token after a character in an unsloped font; we believe (in early 2003) that this is perhaps inefficient but not dangerous.

It also now checks for empty contents of the text command and optimises this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other `aftergroup` tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```

1 (*2kernel)
2 \def \DeclareTextFontCommand #1#2{%
3   \DeclareRobustCommand#1[1]{%
4     \ifmmode
5       \nfss@text{#2##1}%

```

```

6     \else
7     \hmode@bgroup
8     \text@command{##1}%
9     #2\check@icl ##1\check@icr
10    \expandafter
11    \egroup
12    \fi
13    }%
14 }

\textrm Now we define the \text{family} commands in terms of the above; \texttt does
\textsf not look very nice!
\texttt 15 \DeclareTextFontCommand{\textrm}{\rmfamily}
\textnormal 16 \DeclareTextFontCommand{\textsf}{\sffamily}
17 \DeclareTextFontCommand{\texttt}{\ttfamily}
18 \DeclareTextFontCommand{\textnormal}{\normalfont}

\textbf For the series attribute:
\textmd 19 \DeclareTextFontCommand{\textbf}{\bfseries}
20 \DeclareTextFontCommand{\textmd}{\mdseries}

\textit And for the shapes:
\textsl 21 \DeclareTextFontCommand{\textit}{\itshape}
\textsc 22 \DeclareTextFontCommand{\textsl}{\slshape}
\textup 23 \DeclareTextFontCommand{\textsc}{\scshape}
24 \DeclareTextFontCommand{\textup}{\upshape}

textulc
textsw 25 (/2kernel)
textssc 26 (*2kernel | latexrelease)
27 (latexrelease)\IncludeInRelease{2020/02/02}%
28 (latexrelease)      {\textulc}{Additional text commands}%
29 \DeclareTextFontCommand{\textulc}{\ulcshape}
30 \DeclareTextFontCommand{\textsw}{\swshape}
31 \DeclareTextFontCommand{\textssc}{\sscshape}
32 (/2kernel | latexrelease)
33 (latexrelease)\EndIncludeInRelease
34 (latexrelease)\IncludeInRelease{0000/00/00}%
35 (latexrelease)      {\textulc}{Additional text commands}%
36 (latexrelease)
37 (latexrelease)\let\textulc@undefined
38 (latexrelease)\let\textsw@undefined
39 (latexrelease)\let\textssc@undefined
40 (latexrelease)\EndIncludeInRelease
41 (*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:\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   \maybeicfalse
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   \fi
```

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
```

55 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

ltextcomp.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\sfdef@ult
30         \fontfamily\sfsbstdefault
31     \else\ifx\f@family\ttdef@ult
32         \fontfamily\ttsubstdefault
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\symbletters #1%
50 <latexrelease>     \endgroup
51 <latexrelease>}
52 <latexrelease>\let\legacyoldstylenums\@undefined
53 <latexrelease>\def\textcompsubstdefault{cmr}
54 <latexrelease>
55 <latexrelease>\EndIncludeInRelease

```

Everything else in the 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

```
\DeclareEncodingSubset{TS1}{Alegreya-LF}{2}  
\DeclareEncodingSubset{TS1}{Alegreya-OfS}{2}  
\DeclareEncodingSubset{TS1}{Alegreya-TLF}{2}  
\DeclareEncodingSubset{TS1}{Alegreya-TOfS}{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 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}{#2OfS}{#4}%  
71     \DeclareEncodingSubset{#1}{#2TOfS}{#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\textcompstbstdefault
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}\cr
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}
```

56 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

```

Using `\UndeclareTextCommand` above is enough only if the encoding definition files are not reloaded afterwards. In the past that happened if `fontenc` was used in the document preamble (not not any longer). So in some sense it is better to fully remove them from the encoding files, but for rollbacks it is easier to keep them in for now.

These are the standard `itemize` and footnote symbols originally taken from OMS and now from TS1:

```

140 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
141 \DeclareTextSymbolDefault{\textbullet}{TS1}
142 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}

```

```

143 \DeclareTextSymbolDefault{\textdagger}{TS1}
144 \DeclareTextSymbolDefault{\textparagraph}{TS1}
145 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
146 \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.

```

147 %%\DeclareTextSymbolDefault{\textbardbl}{TS1} % subst in sub-enc 9 above
148 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
149 %%\DeclareTextSymbolDefault{\textcelsius}{TS1} % subst in sub-enc 9 above
150 \DeclareTextSymbolDefault{\textcent}{TS1}
151 \DeclareTextSymbolDefault{\textcopyright}{TS1}
152 \DeclareTextSymbolDefault{\textdegree}{TS1}
153 \DeclareTextSymbolDefault{\textdiv}{TS1}
154 \DeclareTextSymbolDefault{\textlnot}{TS1}
155 \DeclareTextSymbolDefault{\textonehalf}{TS1}
156 \DeclareTextSymbolDefault{\textonequarter}{TS1}
157 %%\DeclareTextSymbolDefault{\textonesuperior}{TS1} % subst in sub-enc 9 above
158 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
159 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
160 \DeclareTextSymbolDefault{\textpm}{TS1}
161 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
162 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
163 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
164 \DeclareTextSymbolDefault{\textregistered}{TS1}
165 %%\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1} % subst in sub-enc 9 above
166 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
167 %%\DeclareTextSymbolDefault{\textthreesuperior}{TS1} % subst in sub-enc 9 above
168 \DeclareTextSymbolDefault{\texttimes}{TS1}
169 \DeclareTextSymbolDefault{\texttrademark}{TS1}
170 %%\DeclareTextSymbolDefault{\texttwelveudash}{TS1} % subst in sub-enc 9 above
171 %%\DeclareTextSymbolDefault{\textttwosuperior}{TS1} % subst in sub-enc 9 above
172 \DeclareTextSymbolDefault{\textyen}{TS1}

173 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
174 \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.

56.1 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.

```

175 \DeclareTextCommandDefault{\textcircled}
176 {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OMS}}1}\textcircled}

```

56.2 Sub-encoding 2 (majority of new OTF fonts via autoinst)

```
177 \DeclareTextCommandDefault{\t}
178 {\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.

```
179 \ifx\Umathcode\undefined
180 \DeclareTextCommandDefault{\capitalacut} {\tc@check@accent{\'}2\capitalacut}
181 \DeclareTextCommandDefault{\capitalbreve} {\tc@check@accent{\u}2\capitalbreve}
182 \DeclareTextCommandDefault{\capitalcaron} {\tc@check@accent{\v}2\capitalcaron}
183 \DeclareTextCommandDefault{\capitalcedilla} {\tc@check@accent{\c}2\capitalcedilla}
184 \DeclareTextCommandDefault{\capitalcircumflex} {\tc@check@accent{\^}2\capitalcircumflex}
185 \DeclareTextCommandDefault{\capitaldieresis} {\tc@check@accent{\"}2\capitaldieresis}
186 \DeclareTextCommandDefault{\capitaldotaccent} {\tc@check@accent{\.}2\capitaldotaccent}
187 \DeclareTextCommandDefault{\capitalgrave} {\tc@check@accent{\`}2\capitalgrave}
188 \DeclareTextCommandDefault{\capitalhungarumlaut} {\tc@check@accent{\H}2\capitalhungarumlaut}
189 \DeclareTextCommandDefault{\capitalmacron} {\tc@check@accent{\=}2\capitalmacron}
190 \DeclareTextCommandDefault{\capitalogonek} {\tc@check@accent{\k}2\capitalogonek}
191 \DeclareTextCommandDefault{\capitalring} {\tc@check@accent{\r}2\capitalring}
192 \DeclareTextCommandDefault{\capitaltie} {\tc@check@accent{\t}2\capitaltie}
193 \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).

```
194 \DeclareTextCommandDefault{\newtie} {\tc@check@accent{\t}2\newtie}
195 \DeclareTextCommandDefault{\capitalnewtie} {\tc@check@accent{\t}2\capitalnewtie}
```

In Unicode engines we just execute the simple accents:

```
196 \else
197 \def\capitalacut{\@tabacckludge'}
198 \def\capitalbreve{\u}
199 \def\capitalcaron{\v}
200 \def\capitalcedilla{\c}
201 \def\capitalcircumflex{\^}
202 \def\capitaldieresis{\"}
203 \def\capitaldotaccent{\.}
204 \def\capitalgrave{\@tabacckludge`}
205 \def\capitalhungarumlaut{\H}
206 \def\capitalmacron{\@tabacckludge=}
207 \def\capitalnewtie{\t}
208 \def\capitalogonek{\k}
209 \def\capitalring{\r}
210 \def\capitaltie{\t}
211 \def\capitaltilde{\~}
212 \def\newtie{\t}
213 \fi
```

The next two symbols exist in some fonts (faked?), but we ignore that to keep the subsets reasonable compact and most important linear.

```
214 \DeclareTextCommandDefault{\textlbrackdbl} {\tc@check@symbol2\textlbrackdbl}
215 \DeclareTextCommandDefault{\textrbrackdbl} {\tc@check@symbol2\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.

```

216 \DeclareTextCommandDefault{\texteightoldstyle} {\tc@check@symbol2\texteightoldstyle}
217 \DeclareTextCommandDefault{\textfiveoldstyle} {\tc@check@symbol2\textfiveoldstyle}
218 \DeclareTextCommandDefault{\textfouroldstyle} {\tc@check@symbol2\textfouroldstyle}
219 \DeclareTextCommandDefault{\textnineoldstyle} {\tc@check@symbol2\textnineoldstyle}
220 \DeclareTextCommandDefault{\textoneoldstyle} {\tc@check@symbol2\textoneoldstyle}
221 \DeclareTextCommandDefault{\textsevenoldstyle} {\tc@check@symbol2\textsevenoldstyle}
222 \DeclareTextCommandDefault{\textsixoldstyle} {\tc@check@symbol2\textsixoldstyle}
223 \DeclareTextCommandDefault{\textthreeoldstyle} {\tc@check@symbol2\textthreeoldstyle}
224 \DeclareTextCommandDefault{\texttwooldstyle} {\tc@check@symbol2\texttwooldstyle}
225 \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 autostint, so we pretend there aren't available in sub-encoding 2 and below.

```

226 \DeclareTextCommandDefault{\textacutedbl} {\tc@check@symbol2\textacutedbl}
227 \DeclareTextCommandDefault{\textasciiacute} {\tc@check@symbol2\textasciiacute}
228 \DeclareTextCommandDefault{\textasciibreve} {\tc@check@symbol2\textasciibreve}
229 \DeclareTextCommandDefault{\textasciicaron} {\tc@check@symbol2\textasciicaron}
230 \DeclareTextCommandDefault{\textasciidieresis} {\tc@check@symbol2\textasciidieresis}
231 \DeclareTextCommandDefault{\textasciigrave} {\tc@check@symbol2\textasciigrave}
232 \DeclareTextCommandDefault{\textasciimacron} {\tc@check@symbol2\textasciimacron}
233 \DeclareTextCommandDefault{\textgravedbl} {\tc@check@symbol2\textgravedbl}
234 \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 TeX world.

```

235 \DeclareTextCommandDefault{\textbaht} {\tc@check@symbol2\textbaht}
236 \DeclareTextCommandDefault{\textbigcircle} {\tc@check@symbol2\textbigcircle}
237 \DeclareTextCommandDefault{\textborn} {\tc@check@symbol2\textborn}
238 \DeclareTextCommandDefault{\textcentoldstyle} {\tc@check@symbol2\textcentoldstyle}
239 \DeclareTextCommandDefault{\textcircledP} {\tc@check@symbol2\textcircledP}
240 \DeclareTextCommandDefault{\textcopleft} {\tc@check@symbol2\textcopleft}
241 \DeclareTextCommandDefault{\textdblhyphenchar} {\tc@check@symbol2\textdblhyphenchar}
242 \DeclareTextCommandDefault{\textdblhyphen} {\tc@check@symbol2\textdblhyphen}
243 \DeclareTextCommandDefault{\textdied} {\tc@check@symbol2\textdied}
244 \DeclareTextCommandDefault{\textdiscount} {\tc@check@symbol2\textdiscount}
245 \DeclareTextCommandDefault{\textdivorced} {\tc@check@symbol2\textdivorced}
246 \DeclareTextCommandDefault{\textdollaroldstyle} {\tc@check@symbol2\textdollaroldstyle}
247 \DeclareTextCommandDefault{\textguarani} {\tc@check@symbol2\textguarani}
248 \DeclareTextCommandDefault{\textleaf} {\tc@check@symbol2\textleaf}
249 \DeclareTextCommandDefault{\textlquill} {\tc@check@symbol2\textlquill}
250 \DeclareTextCommandDefault{\textmarried} {\tc@check@symbol2\textmarried}
251 \DeclareTextCommandDefault{\textmho} {\tc@check@symbol2\textmho}
252 \DeclareTextCommandDefault{\textmusicalnote} {\tc@check@symbol2\textmusicalnote}
253 \DeclareTextCommandDefault{\textnaira} {\tc@check@symbol2\textnaira}
254 \DeclareTextCommandDefault{\textopenbullet} {\tc@check@symbol2\textopenbullet}
255 \DeclareTextCommandDefault{\textpeso} {\tc@check@symbol2\textpeso}
256 \DeclareTextCommandDefault{\textpilcrow} {\tc@check@symbol2\textpilcrow}
257 \DeclareTextCommandDefault{\textrecipe} {\tc@check@symbol2\textrecipe}
258 \DeclareTextCommandDefault{\textreferencemark} {\tc@check@symbol2\textreferencemark}
259 \DeclareTextCommandDefault{\textrquill} {\tc@check@symbol2\textrquill}

```

```
260 \DeclareTextCommandDefault{\textservicemark} {\tc@check@symbol2\textservicemark}
261 \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.

```
262 \DeclareTextCommandDefault{\textpertenthousand} {\tc@check@symbol2\textpertenthousand}
263 \UndeclareTextCommand{\textpertenthousand}{T1}
```

56.3 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.

```
264 \DeclareTextCommandDefault{\textlangle} {\tc@check@symbol3\textlangle}
265 \DeclareTextCommandDefault{\textrangle} {\tc@check@symbol3\textrangle}
```

56.4 Sub-encoding 4

```
266 \DeclareTextCommandDefault{\textcolonmonetary} {\tc@check@symbol4\textcolonmonetary}
267 \DeclareTextCommandDefault{\textdong} {\tc@check@symbol4\textdong}
268 \DeclareTextCommandDefault{\textdownarrow} {\tc@check@symbol4\textdownarrow}
269 \DeclareTextCommandDefault{\textleftarrow} {\tc@check@symbol4\textleftarrow}
270 \DeclareTextCommandDefault{\textlira} {\tc@check@symbol4\textlira}
271 \DeclareTextCommandDefault{\textrightarrow} {\tc@check@symbol4\textrightarrow}
272 \DeclareTextCommandDefault{\textuparrow} {\tc@check@symbol4\textuparrow}
273 \DeclareTextCommandDefault{\textwon} {\tc@check@symbol4\textwon}
```

56.5 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.

```
274 \DeclareTextCommandDefault{\textestimated} {\tc@check@symbol5\textestimated}
275 \DeclareTextCommandDefault{\textnumero} {\tc@check@symbol5\textnumero}
```

56.6 Sub-encoding 6

```
276 \DeclareTextCommandDefault{\textflorin} {\tc@check@symbol6\textflorin}
277 \DeclareTextCommandDefault{\textcurrency} {\tc@check@symbol6\textcurrency}
```

56.7 Sub-encoding 7

```
278 \DeclareTextCommandDefault{\textfractionsolidus}{\tc@check@symbol7\textfractionsolidus}
279 \DeclareTextCommandDefault{\textohm} {\tc@check@symbol7\textohm}
```

```
280 \DeclareTextCommandDefault{\textmu}          {\tc@check@symbol7\textmu}
281 \DeclareTextCommandDefault{\textminus}      {\tc@check@symbol7\textminus}
```

56.8 Sub-encoding 8

```
282 \DeclareTextCommandDefault{\textblank}      {\tc@check@symbol8\textblank}
283 \DeclareTextCommandDefault{\textinterrobangdown}{\tc@check@symbol8\textinterrobangdown}
284 \DeclareTextCommandDefault{\textinterrobang} {\tc@check@symbol8\textinterrobang}
```

Fonts with this sub-encoding don't have a Euro symbol, but instead of substituting we fake it.

```
285 \DeclareTextCommandDefault{\texteuro}
286      {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro{8}\texteuro}
```

56.9 Sub-encoding 9 (most missing)

```
287 \DeclareTextCommandDefault{\textcelsius}{\tc@check@symbol9\textcelsius}
288 \DeclareTextCommandDefault{\textonesuperior}{\tc@check@symbol9\textonesuperior}
289 \DeclareTextCommandDefault{\textthreequartersemdash}{\tc@check@symbol9\textthreequartersemdash}
290 \DeclareTextCommandDefault{\textthreesuperior}{\tc@check@symbol9\textthreesuperior}
291 \DeclareTextCommandDefault{\texttwelveudash}{\tc@check@symbol9\texttwelveudash}
292 \DeclareTextCommandDefault{\texttwosuperior}{\tc@check@symbol9\texttwosuperior}
293 \DeclareTextCommandDefault{\textbardbl}{\tc@check@symbol9\textbardbl}
```

57 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).

```
294 \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.

```
295 %\DeclareTextSymbol{\textcopyleft}{TS1}{171}
296 %\DeclareTextSymbol{\textdblhyphen}{TS1}{45}
297 %\DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}
298 %\DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
299 %\DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
300 %\DeclareTextSymbol{\textleaf}{TS1}{108}
301 %\DeclareTextSymbol{\texttwelveudash}{TS1}{21}
302 %\DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
```

If oldstyle numerals are asked for we just use `\oldstylenums`.

```
303 \DeclareTextCommand{\textzerooldstyle} \UnicodeEncodingName{\oldstylenums{0}}
304 \DeclareTextCommand{\textoneoldstyle} \UnicodeEncodingName{\oldstylenums{1}}
305 \DeclareTextCommand{\texttwooldstyle} \UnicodeEncodingName{\oldstylenums{2}}
306 \DeclareTextCommand{\textthreeoldstyle} \UnicodeEncodingName{\oldstylenums{3}}
307 \DeclareTextCommand{\textfouroldstyle} \UnicodeEncodingName{\oldstylenums{4}}
308 \DeclareTextCommand{\textfiveoldstyle} \UnicodeEncodingName{\oldstylenums{5}}
309 \DeclareTextCommand{\textsixoldstyle} \UnicodeEncodingName{\oldstylenums{6}}
310 \DeclareTextCommand{\textsevenoldstyle} \UnicodeEncodingName{\oldstylenums{7}}
311 \DeclareTextCommand{\texteightoldstyle} \UnicodeEncodingName{\oldstylenums{8}}
312 \DeclareTextCommand{\textnineoldstyle} \UnicodeEncodingName{\oldstylenums{9}}
```

These have Unicode slots so this should be integrated into TU explicitly

```
313 \DeclareTextSymbol{\textpilcrow}          \UnicodeEncodingName{"00B6}
314 \DeclareTextSymbol{\textborn}            \UnicodeEncodingName{"002A}
```

```

315 \DeclareTextSymbol{\textdied}          \UnicodeEncodingName{"2020}
316 \DeclareTextSymbol{\textlbrackdbl}     \UnicodeEncodingName{"27E6}
317 \DeclareTextSymbol{\textrbrackdbl}     \UnicodeEncodingName{"27E7}
318 \DeclareTextSymbol{\textguarani}       \UnicodeEncodingName{"20B2}

We could make \textcentoldstyle and \textdollaroldstyle point to dollar
and cent in the Unicode encoding

319 %\DeclareTextSymbol{\textcentoldstyle} \UnicodeEncodingName{"00A2}
320 %\DeclareTextSymbol{\textdollaroldstyle} \UnicodeEncodingName{"0024}

but I think it is better to pick them up from TS1 even if that usually means LMR
fonts

321 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
322 \DeclareTextSymbol{\textcentoldstyle} {TS1}{139}

323 \fi          % --- END of Unicode engines specials

```

58 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 likely much too conservative, but with the advantage that we aren't getting missing glyphs (or at least that this is 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.

```
324 \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.

```

325 \DeclareEncodingSubset{TS1}{ccr}      {0}
326 \DeclareEncodingSubset{TS1}{cmbr}    {0}
327 \DeclareEncodingSubset{TS1}{cmr}     {0}
328 \DeclareEncodingSubset{TS1}{cmss}    {0}
329 \DeclareEncodingSubset{TS1}{cmtl}    {0}
330 \DeclareEncodingSubset{TS1}{cmtt}    {0}
331 \DeclareEncodingSubset{TS1}{cmvtt}   {0}
332 \DeclareEncodingSubset{TS1}{pxr}     {0}
333 \DeclareEncodingSubset{TS1}{pxss}    {0}
334 \DeclareEncodingSubset{TS1}{pxtt}    {0}
335 \DeclareEncodingSubset{TS1}{qag}     {0}
336 \DeclareEncodingSubset{TS1}{qbk}     {0}
337 \DeclareEncodingSubset{TS1}{qcr}     {0}
338 \DeclareEncodingSubset{TS1}{qcs}     {0}
339 \DeclareEncodingSubset{TS1}{qhvc}    {0}
340 \DeclareEncodingSubset{TS1}{qhv}     {0}
341 \DeclareEncodingSubset{TS1}{qpl}     {0}
342 \DeclareEncodingSubset{TS1}{qtm}     {0}
343 \DeclareEncodingSubset{TS1}{qzc}     {0}
344 \DeclareEncodingSubset{TS1}{txr}     {0}
345 \DeclareEncodingSubset{TS1}{txss}    {0}
346 \DeclareEncodingSubset{TS1}{txtt}    {0}

```

```

347 \DeclareEncodingSubset{TS1}{lmr}      {1}
348 \DeclareEncodingSubset{TS1}{lmdh}     {1}
349 \DeclareEncodingSubset{TS1}{lmss}     {1}
350 \DeclareEncodingSubset{TS1}{lmssq}    {1}
351 \DeclareEncodingSubset{TS1}{lmvtt}    {1}
352 \DeclareEncodingSubset{TS1}{lmtt}     {1} % missing TM, SM, pertenthousand for some reason

353 \DeclareEncodingSubset{TS1}{ptmx}     {2}
354 \DeclareEncodingSubset{TS1}{ptmj}     {2}
355 \DeclareEncodingSubset{TS1}{ul8}      {2}

356 \DeclareEncodingSubset{TS1}{bch}      {5} % tofu for blank, ohm
357 \DeclareEncodingSubset{TS1}{futj}     {5} % tofu for blank, interrobang/down, ohm
358 \DeclareEncodingSubset{TS1}{futs}     {5} % tofu for blank, ohm
359 \DeclareEncodingSubset{TS1}{futex}    {5} % probably (currently broken distrib)
360 \DeclareEncodingSubset{TS1}{pag}      {5} % tofu for blank, interrobang/down, ohm
361 \DeclareEncodingSubset{TS1}{pbk}      {5} % tofu for blank, interrobang/down, ohm
362 \DeclareEncodingSubset{TS1}{pcr}      {5} % tofu for blank, interrobang/down, ohm
363 \DeclareEncodingSubset{TS1}{phv}      {5} % tofu for blank, interrobang/down, ohm
364 \DeclareEncodingSubset{TS1}{pnc}      {5} % tofu for blank, interrobang/down, ohm
365 \DeclareEncodingSubset{TS1}{pplj}     {5} % tofu for blank
366 \DeclareEncodingSubset{TS1}{pplx}     {5} % tofu for blank
367 \DeclareEncodingSubset{TS1}{ppl}      {5} % tofu for blank interrobang/down
368 \DeclareEncodingSubset{TS1}{ptm}      {5} % tofu for blank, interrobang/down, ohm
369 \DeclareEncodingSubset{TS1}{pzc}      {5} % tofu for blank, interrobang/down, ohm
370 \DeclareEncodingSubset{TS1}{ul9}      {5} % tofu for blank, interrobang/down, ohm

371 \DeclareEncodingSubset{TS1}{dayroms}  {6} % tofu for blank, interrobang/down, ohm
372 \DeclareEncodingSubset{TS1}{dayrom}   {6} % tofu for blank, interrobang/down, ohm

373 \DeclareEncodingSubset{TS1}{augie}    {8} % really only missing euro
374 \DeclareEncodingSubset{TS1}{put}      {8}
375 \DeclareEncodingSubset{TS1}{uag}      {8} % probably (currently broken distrib)
376 \DeclareEncodingSubset{TS1}{ugq}      {8}

377 \DeclareEncodingSubset{TS1}{zi4}      {9}

```

LucidaBright (sold through TUG) probably not quite correct, I guess as I have the older fonts ...

```

378 \DeclareEncodingSubset{TS1}{hls}      {5}
379 \DeclareEncodingSubset{TS1}{hlst}     {5}
380 \DeclareEncodingSubset{TS1}{hlct}     {5}
381 \DeclareEncodingSubset{TS1}{hlh}      {5}
382 \DeclareEncodingSubset{TS1}{hlx}      {8}
383 \DeclareEncodingSubset{TS1}{hlce}     {8}
384 \DeclareEncodingSubset{TS1}{hlcn}     {8}
385 \DeclareEncodingSubset{TS1}{hlcw}     {8}
386 \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 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).

```

387 \DeclareEncodingSubset{TS1}{lato-*}   {0} % with a bunch of tofu inside

```

```

388 \DeclareEncodingSubset{TS1}{opensans-*}           {0} % with a bunch of tofu inside
389 \DeclareEncodingSubset{TS1}{cantarell-*}          {0} % with a bunch of tofu inside
390 \DeclareEncodingSubset{TS1}{fbb-*}                {0} % missing centoldstyle
391 \DeclareEncodingSubset{TS1}{tli}                  {1} % with lots of tofu inside
392 \DeclareEncodingSubset{TS1}{Alegreya-*}           {2}
393 \DeclareEncodingSubset{TS1}{AlegreyaSans-*}       {2}
394 \DeclareEncodingSubset{TS1}{DejaVuSans-TLF}       {2}
395 \DeclareEncodingSubset{TS1}{DejaVuSansCondensed-TLF} {2}
396 \DeclareEncodingSubset{TS1}{DejaVuSansMono-TLF}  {2}
397 \DeclareEncodingSubset{TS1}{EBGaramond-*}        {2}
398 \DeclareEncodingSubset{TS1}{Tempora-TLF}         {2}
399 \DeclareEncodingSubset{TS1}{Tempora-TOfF}        {2}

400 \DeclareEncodingSubset{TS1}{Arimo-TLF}           {3}
401 \DeclareEncodingSubset{TS1}{Carlito-*}           {3}
402 \DeclareEncodingSubset{TS1}{FiraSans-*}          {3}
403 \DeclareEncodingSubset{TS1}{IBMPlexSans-TLF}     {3}
404 \DeclareEncodingSubset{TS1}{Merriweather-OfF}    {3}
405 \DeclareEncodingSubset{TS1}{Montserrat-*}        {3}
406 \DeclareEncodingSubset{TS1}{MontserratAlternates-*} {3}
407 \DeclareEncodingSubset{TS1}{SourceCodePro-TLF}   {3}
408 \DeclareEncodingSubset{TS1}{SourceCodePro-TOfF}  {3}
409 \DeclareEncodingSubset{TS1}{SourceSansPro-*}     {3}
410 \DeclareEncodingSubset{TS1}{SourceSerifPro-*}    {3}
411 \DeclareEncodingSubset{TS1}{Tinos-TLF}           {3}

412 \DeclareEncodingSubset{TS1}{AccanthisADFStdNoThree-LF} {4}
413 \DeclareEncodingSubset{TS1}{Cabin-TLF}           {4}
414 \DeclareEncodingSubset{TS1}{Caladea-TLF}         {4}
415 \DeclareEncodingSubset{TS1}{Chivo-*}             {4}
416 \DeclareEncodingSubset{TS1}{ClearSans-TLF}       {4}
417 \DeclareEncodingSubset{TS1}{Coelacanth-LF}      {4}
418 \DeclareEncodingSubset{TS1}{CrimsonPro-*}       {4}
419 \DeclareEncodingSubset{TS1}{FiraMono-TLF}        {4}
420 \DeclareEncodingSubset{TS1}{FiraMono-TOfF}      {4}
421 \DeclareEncodingSubset{TS1}{Go-TLF}              {4}
422 \DeclareEncodingSubset{TS1}{GoMono-TLF}         {4}
423 \DeclareEncodingSubset{TS1}{InriaSans-*}         {4}
424 \DeclareEncodingSubset{TS1}{InriaSerif-*}        {4}
425 \DeclareEncodingSubset{TS1}{LibertinusSans-*}    {4}
426 \DeclareEncodingSubset{TS1}{LibertinusSerif-*}   {4}
427 \DeclareEncodingSubset{TS1}{LibreBodoni-TLF}    {4}
428 \DeclareEncodingSubset{TS1}{LibreFranklin-TLF}  {4}
429 \DeclareEncodingSubset{TS1}{LinguisticsPro-LF}  {4}
430 \DeclareEncodingSubset{TS1}{LinguisticsPro-OfF} {4}
431 \DeclareEncodingSubset{TS1}{LinuxBioluminumT-*}  {4}
432 \DeclareEncodingSubset{TS1}{LinuxLibertineT-*}  {4}
433 \DeclareEncodingSubset{TS1}{MerriweatherSans-OfF} {4}
434 \DeclareEncodingSubset{TS1}{MintSpirit-*}        {4}
435 \DeclareEncodingSubset{TS1}{MintSpiritNoTwo-*}   {4}
436 \DeclareEncodingSubset{TS1}{PTMono-TLF}         {4}
437 \DeclareEncodingSubset{TS1}{PTSans-TLF}         {4}
438 \DeclareEncodingSubset{TS1}{PTSansCaption-TLF}  {4}
439 \DeclareEncodingSubset{TS1}{PTSansNarrow-TLF}   {4}
440 \DeclareEncodingSubset{TS1}{PTSerif-TLF}        {4}

```

441 \DeclareEncodingSubset{TS1}{PTSerifCaption-TLF} {4}
442 \DeclareEncodingSubset{TS1}{Raleway-TLF} {4}
443 \DeclareEncodingSubset{TS1}{Raleway-TOfF} {4}
444 \DeclareEncodingSubset{TS1}{Roboto-*} {4}
445 \DeclareEncodingSubset{TS1}{RobotoMono-TLF} {4}
446 \DeclareEncodingSubset{TS1}{RobotoSlab-TLF} {4}
447 \DeclareEncodingSubset{TS1}{Rosario-*} {4}
448 \DeclareEncodingSubset{TS1}{SticksTooText-*} {4}
449 \DeclareEncodingSubset{TS1}{UniversalisADFStd-LF} {4}

450 \DeclareEncodingSubset{TS1}{Almendra-OfF} {5}
451 \DeclareEncodingSubset{TS1}{Baskervaldx-*} {5}
452 \DeclareEncodingSubset{TS1}{BaskervilleF-*} {5}
453 \DeclareEncodingSubset{TS1}{Bitter-TLF} {5}
454 \DeclareEncodingSubset{TS1}{Cinzel-LF} {5}
455 \DeclareEncodingSubset{TS1}{CinzelDecorative-LF} {5}
456 \DeclareEncodingSubset{TS1}{DejaVuSerif-TLF} {5}
457 \DeclareEncodingSubset{TS1}{DejaVuSerifCondensed-TLF} {5}
458 \DeclareEncodingSubset{TS1}{GilliusADF-LF} {5}
459 \DeclareEncodingSubset{TS1}{GilliusADFCond-LF} {5}
460 \DeclareEncodingSubset{TS1}{GilliusADFNoTwo-LF} {5}
461 \DeclareEncodingSubset{TS1}{GilliusADFNoTwoCond-LF} {5}
462 \DeclareEncodingSubset{TS1}{LobsterTwo-LF} {5}
463 \DeclareEncodingSubset{TS1}{OldStandard-TLF} {5}
464 \DeclareEncodingSubset{TS1}{PlayfairDisplay-TLF} {5}
465 \DeclareEncodingSubset{TS1}{PlayfairDisplay-TOfF} {5}
466 \DeclareEncodingSubset{TS1}{TheanoDidot-TLF} {5}
467 \DeclareEncodingSubset{TS1}{TheanoDidot-TOfF} {5}
468 \DeclareEncodingSubset{TS1}{TheanoModern-TLF} {5}
469 \DeclareEncodingSubset{TS1}{TheanoModern-TOfF} {5}
470 \DeclareEncodingSubset{TS1}{TheanoOldStyle-TLF} {5}
471 \DeclareEncodingSubset{TS1}{TheanoOldStyle-TOfF} {5}

472 \DeclareEncodingSubset{TS1}{Crimson-TLF} {6}
473 \DeclareEncodingSubset{TS1}{IBMPlexMono-TLF} {6}
474 \DeclareEncodingSubset{TS1}{IBMPlexSerif-TLF} {6}
475 \DeclareEncodingSubset{TS1}{LibertinusMono-TLF} {6}
476 \DeclareEncodingSubset{TS1}{LibertinusSerifDisplay-LF} {6}
477 \DeclareEncodingSubset{TS1}{LinuxLibertineDisplayT-*} {6}
478 \DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-LF} {6}
479 \DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-TLF} {6}
480 \DeclareEncodingSubset{TS1}{Overlock-LF} {6}

481 \DeclareEncodingSubset{TS1}{CormorantGaramond-*} {7}
482 \DeclareEncodingSubset{TS1}{Heuristica-TLF} {7}
483 \DeclareEncodingSubset{TS1}{Heuristica-TOfF} {7}
484 \DeclareEncodingSubset{TS1}{IMFELLEnglish-TLF} {7}
485 \DeclareEncodingSubset{TS1}{LibreBaskerville-TLF} {7}
486 \DeclareEncodingSubset{TS1}{LibreCaslon-*} {7}
487 \DeclareEncodingSubset{TS1}{Marcellus-LF} {7}
488 \DeclareEncodingSubset{TS1}{NotoSans-*} {7}
489 \DeclareEncodingSubset{TS1}{NotoSansMono-TLF} {7}
490 \DeclareEncodingSubset{TS1}{NotoSansMono-TOfF} {7}
491 \DeclareEncodingSubset{TS1}{NotoSerif-*} {7}
492 \DeclareEncodingSubset{TS1}{Quattrocento-TLF} {7}
493 \DeclareEncodingSubset{TS1}{QuattrocentoSans-TLF} {7}

```

494 \DeclareEncodingSubset{TS1}{XCharter-TLF}           {7}
495 \DeclareEncodingSubset{TS1}{XCharter-TOfS}         {7}
496 \DeclareEncodingSubset{TS1}{erewhon-*}            {7}
497 \DeclareEncodingSubset{TS1}{ComicNeue-TLF}        {7}
498 \DeclareEncodingSubset{TS1}{ComicNeueAngular-TLF} {7}
499 \DeclareEncodingSubset{TS1}{Forum-LF}             {7} % the superiors are missing
500 \DeclareEncodingSubset{TS1}{Cochineal-*}          {8}
501 \DeclareEncodingSubset{TS1}{AlgolRevived-TLF}     {9}

```

59 Legacy symbol support for lists and footnote symbols

`\UseLegacyTextSymbols`

```

502 \def\UseLegacyTextSymbols{%
503   \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}%
504   \DeclareTextSymbolDefault{\textbardbl}{OMS}%
505   \DeclareTextSymbolDefault{\textbullet}{OMS}%
506   \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}%
507   \DeclareTextSymbolDefault{\textdagger}{OMS}%
508   \DeclareTextSymbolDefault{\textparagraph}{OMS}%
509   \DeclareTextSymbolDefault{\textperiodcentered}{OMS}%
510   \DeclareTextSymbolDefault{\textsection}{OMS}%
511   \UndeclareTextCommand{\textsection}{T1}%
512   \expandafter\let\csname oldstylenums \expandafter\endcsname
513     \csname legacyoldstylenums \endcsname
514 }

```

`\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`

```

515 \DeclareTextSymbol{\textlegacyasteriskcentered}{OMS}{3} % "03
516 \DeclareTextSymbol{\textlegacybardbl}{OMS}{107} % "6B
517 \DeclareTextSymbol{\textlegacybullet}{OMS}{15} % "0F
518 \DeclareTextSymbol{\textlegacydaggerdbl}{OMS}{122} % "7A
519 \DeclareTextSymbol{\textlegacydagger}{OMS}{121} % "79
520 \DeclareTextSymbol{\textlegacyparagraph}{OMS}{123} % "7B
521 \DeclareTextSymbol{\textlegacyperiodcentered}{OMS}{1} % "01
522 \DeclareTextSymbol{\textlegacysection}{OMS}{120} % "78

```

```

523 \DeclareTextSymbolDefault{\textlegacyasteriskcentered}{OMS}
524 \DeclareTextSymbolDefault{\textlegacybardbl}{OMS}
525 \DeclareTextSymbolDefault{\textlegacybullet}{OMS}
526 \DeclareTextSymbolDefault{\textlegacydaggerdbl}{OMS}
527 \DeclareTextSymbolDefault{\textlegacydagger}{OMS}
528 \DeclareTextSymbolDefault{\textlegacyparagraph}{OMS}
529 \DeclareTextSymbolDefault{\textlegacyperiodcentered}{OMS}
530 \DeclareTextSymbolDefault{\textlegacysection}{OMS}

```

Supporting rollback ...

```
531 </2ekernel | latexrelease>
532 <latexrelease>\EndIncludeInRelease
533 <latexrelease>\IncludeInRelease{0000/00/00}%
534 <latexrelease>  {\DeclareEncodingSubset}{Text companion symbols}%
535 <latexrelease>
536 <latexrelease>\let\DeclareEncodingSubset\@undefined
537 <latexrelease>\let\CheckEncodingSubset\@undefined
538 <latexrelease>
539 <latexrelease>\DeclareTextSymbolDefault{\textdollar}{OT1}
540 <latexrelease>\DeclareTextSymbolDefault{\textsterling}{OT1}
541 <latexrelease>\DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
542 <latexrelease>  \ifdim \fontdimen\@ne\font >\z@
543 <latexrelease>    \slshape
544 <latexrelease>  \else
545 <latexrelease>    \upshape
546 <latexrelease>  \fi
547 <latexrelease>  \char'\$\egroup}
548 <latexrelease>\DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
549 <latexrelease>  \ifdim \fontdimen\@ne\font >\z@
550 <latexrelease>    \itshape
551 <latexrelease>  \else
552 <latexrelease>    \fontshape{ui}\selectfont
553 <latexrelease>  \fi
554 <latexrelease>  \char'\$\egroup}
555 <latexrelease>\DeclareTextCommand{\textperthousand}{T1}
556 <latexrelease>  {\%\char 24 }
557 <latexrelease>
558 <latexrelease>\DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
559 <latexrelease>\DeclareTextSymbolDefault{\textbullet}{OMS}
560 <latexrelease>\DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
561 <latexrelease>\DeclareTextSymbolDefault{\textdagger}{OMS}
562 <latexrelease>\DeclareTextSymbolDefault{\textparagraph}{OMS}
563 <latexrelease>\DeclareTextSymbolDefault{\textperiodcentered}{OMS}
564 <latexrelease>\DeclareTextSymbolDefault{\textsection}{OMS}
565 <latexrelease>
566 <latexrelease>\DeclareTextSymbolDefault{\textbardbl}{OMS}
567 <latexrelease>\let\textbrokenbar\@undefined
568 <latexrelease>\let\textcelsius\@undefined
569 <latexrelease>\let\textcent\@undefined
570 <latexrelease>\DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
571 <latexrelease>\let\textdegree\@undefined
572 <latexrelease>\let\textdiv\@undefined
573 <latexrelease>\let\textlnot\@undefined
574 <latexrelease>\let\textonehalf\@undefined
575 <latexrelease>\let\textonequarter\@undefined
576 <latexrelease>\let\textonesuperior\@undefined
577 <latexrelease>\DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
578 <latexrelease>\DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}
579 <latexrelease>\let\textpm\@undefined
580 <latexrelease>\let\textquotesingle\@undefined
581 <latexrelease>\let\textquotestraightbase\@undefined
582 <latexrelease>\let\textquotestraightdblbase\@undefined
583 <latexrelease>\DeclareTextCommandDefault{\textregistered}{\textcircled{R}}
```

```

584 (latexrelease) \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
585 (latexrelease)\let\textthreequartersemdash\@undefined
586 (latexrelease)\let\textthreequarters\@undefined
587 (latexrelease)\let\textthreesuperior\@undefined
588 (latexrelease)\let\texttimes\@undefined
589 (latexrelease)\DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
590 (latexrelease)\let\texttwelveudash\@undefined
591 (latexrelease)\let\texttwosuperior\@undefined
592 (latexrelease)\let\textyen\@undefined
593 (latexrelease)
594 (latexrelease)\let\textcapitalcompwordmark\@undefined
595 (latexrelease)\let\textascendercompwordmark\@undefined
596 (latexrelease)
597 (latexrelease)\DeclareTextAccentDefault{\textcircled}{OMS}
598 (latexrelease)\DeclareTextAccentDefault{\t}{OML}
599 (latexrelease)
600 (latexrelease)\let\capitalacute\@undefined
601 (latexrelease)\let\capitalbreve\@undefined
602 (latexrelease)\let\capitalcaron\@undefined
603 (latexrelease)\let\capitalcedilla\@undefined
604 (latexrelease)\let\capitalcircumflex\@undefined
605 (latexrelease)\let\capitaldieresis\@undefined
606 (latexrelease)\let\capitaldotaccent\@undefined
607 (latexrelease)\let\capitalgrave\@undefined
608 (latexrelease)\let\capitalhungarumlaut\@undefined
609 (latexrelease)\let\capitalmacron\@undefined
610 (latexrelease)\let\capitalnewtie\@undefined
611 (latexrelease)\let\capitalogonek\@undefined
612 (latexrelease)\let\capitalring\@undefined
613 (latexrelease)\let\capitaltie\@undefined
614 (latexrelease)\let\capitaltilde\@undefined
615 (latexrelease)\let\newtie\@undefined
616 (latexrelease)
617 (latexrelease)\let\textlbrackdbl\@undefined
618 (latexrelease)\let\textrbrackdbl\@undefined
619 (latexrelease)
620 (latexrelease)\let\texteightoldstyle\@undefined
621 (latexrelease)\let\textfiveoldstyle\@undefined
622 (latexrelease)\let\textfouroldstyle\@undefined
623 (latexrelease)\let\textnineoldstyle\@undefined
624 (latexrelease)\let\textoneoldstyle\@undefined
625 (latexrelease)\let\textsevenoldstyle\@undefined
626 (latexrelease)\let\textsixoldstyle\@undefined
627 (latexrelease)\let\textthreeoldstyle\@undefined
628 (latexrelease)\let\texttwooldstyle\@undefined
629 (latexrelease)\let\textzerooldstyle\@undefined
630 (latexrelease)
631 (latexrelease)\let\textacutedbl\@undefined
632 (latexrelease)\let\textasciiacute\@undefined
633 (latexrelease)\let\textasciibreve\@undefined
634 (latexrelease)\let\textasciicaron\@undefined
635 (latexrelease)\let\textasciidieresis\@undefined
636 (latexrelease)\let\textasciigrave\@undefined
637 (latexrelease)\let\textasciimacron\@undefined

```

638 (latexrelease)\let\textgravedbl\@undefined
639 (latexrelease)\let\texttildelow\@undefined
640 (latexrelease)
641 (latexrelease)\let\textbaht\@undefined
642 (latexrelease)\let\textbigcircle\@undefined
643 (latexrelease)\let\textborn\@undefined
644 (latexrelease)\let\textcentoldstyle\@undefined
645 (latexrelease)\let\textcircledP\@undefined
646 (latexrelease)\let\textcopyleft\@undefined
647 (latexrelease)\let\textdblhyphenchar\@undefined
648 (latexrelease)\let\textdblhyphen\@undefined
649 (latexrelease)\let\textdied\@undefined
650 (latexrelease)\let\textdiscount\@undefined
651 (latexrelease)\let\textdivorced\@undefined
652 (latexrelease)\let\textdollaroldstyle\@undefined
653 (latexrelease)\let\textguarani\@undefined
654 (latexrelease)\let\textleaf\@undefined
655 (latexrelease)\let\textlquill\@undefined
656 (latexrelease)\let\textmarried\@undefined
657 (latexrelease)\let\textmho\@undefined
658 (latexrelease)\let\textmusicalnote\@undefined
659 (latexrelease)\let\textnaira\@undefined
660 (latexrelease)\let\textopenbullet\@undefined
661 (latexrelease)\let\textpeso\@undefined
662 (latexrelease)\let\textpilcrow\@undefined
663 (latexrelease)\let\textrecipe\@undefined
664 (latexrelease)\let\textreferencemark\@undefined
665 (latexrelease)\let\textrquill\@undefined
666 (latexrelease)\let\textservicemark\@undefined
667 (latexrelease)\let\textsurd\@undefined
668 (latexrelease)
669 (latexrelease)\DeclareTextCommand{\textpertenthousand}{T1}
670 (latexrelease) \{\%\char 24\char 24 }
671 (latexrelease)
672 (latexrelease)\let\textlangle\@undefined
673 (latexrelease)\let\extrangle\@undefined
674 (latexrelease)
675 (latexrelease)\let\textcolonmonetary\@undefined
676 (latexrelease)\let\textdong\@undefined
677 (latexrelease)\let\textdownarrow\@undefined
678 (latexrelease)\let\textleftarrow\@undefined
679 (latexrelease)\let\textlira\@undefined
680 (latexrelease)\let\textrightarrow\@undefined
681 (latexrelease)\let\textuparrow\@undefined
682 (latexrelease)\let\textwon\@undefined
683 (latexrelease)
684 (latexrelease)\let\textestimated\@undefined
685 (latexrelease)\let\textnumero\@undefined
686 (latexrelease)
687 (latexrelease)\let\textflorin\@undefined
688 (latexrelease)\let\textcurrency\@undefined
689 (latexrelease)
690 (latexrelease)\let\textfractionsolidus\@undefined
691 (latexrelease)\let\textohm\@undefined

```

692 (latexrelease)\let\textmu\@undefined
693 (latexrelease)\let\textminus\@undefined
694 (latexrelease)
695 (latexrelease)\let\textblank\@undefined
696 (latexrelease)\let\textinterrobangdown\@undefined
697 (latexrelease)\let\textinterrobang\@undefined
698 (latexrelease)
699 (latexrelease)\let\texteuro\@undefined
700 (latexrelease)
701 (latexrelease)\let\textcelsius\@undefined
702 (latexrelease)\let\textonesuperior\@undefined
703 (latexrelease)\let\textthreequartersemdash\@undefined
704 (latexrelease)\let\textthreesuperior\@undefined
705 (latexrelease)\let\texttwelveudash\@undefined
706 (latexrelease)\let\texttwosuperior\@undefined
707 (latexrelease)\let\textbardbl\@undefined
708 (latexrelease)
709 (latexrelease)\let\UseLegacyTextSymbols\@undefined
710 (latexrelease)\let\textlegacyasteriskcentered\@undefined
711 (latexrelease)\let\textlegacybardbl\@undefined
712 (latexrelease)\let\textlegacybullet\@undefined
713 (latexrelease)\let\textlegacydaggerdbl\@undefined
714 (latexrelease)\let\textlegacydagger\@undefined
715 (latexrelease)\let\textlegacyparagraph\@undefined
716 (latexrelease)\let\textlegacyperiodcentered\@undefined
717 (latexrelease)\let\textlegacysection\@undefined
718 (latexrelease)
719 (latexrelease)\EndIncludeInRelease
720 (*2kernel)
721 (/2kernel)

```

60 The textcomp package

```

722 (*TS1sty)
723 \providecommand\DeclareRelease[3]{
724 \providecommand\DeclareCurrentRelease[2]{
725
726 \DeclareRelease{}{2018-08-11}{textcomp-2018-08-11.sty}
727 \DeclareCurrentRelease{}{2020-02-02}
728
729 \ProvidesPackage{textcomp}
730 [2020/02/02 v2.0n Standard LaTeX package]

```

A precaution in case this is used without rebuilding the format.

```
731 \NeedsTeXFormat{LaTeX2e}[2020/02/02]
```

This is implemented by defining the default subset:

```

732 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
733 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
734 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{8}}
735 \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). Using the package options you can change this behavior.

```
736 \DeclareOption{error}{\gdef\tc@errorwarn{\PackageError{textcomp}}}
737 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2{\PackageWarning{textcomp}{#1}}}
738 \DeclareOption{info}{\gdef\tc@errorwarn#1#2{\PackageInfo{textcomp}{#1}}}
739 \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.

```
740 \DeclareOption{force}{%
741   \def\CheckEncodingSubset#1#2#3#4#5{%
742     \ifnum #4>%
743       0\csname #2:?\endcsname
744       \relax
745     \expandafter\@firstoftwo
746     \else
747     \expandafter\@secondoftwo
748     \fi
749     {#1{#2}}{#3}%
750     #5}%
751 }
752 \ExecuteOptions{info}
753 \ProcessOptions\relax
```

There is not much else to do nowadays, because everything is already set up in the L^AT_EX kernel.

```
754 \InputIfFileExists{textcomp.cfg}
755 {\PackageInfo{textcomp}{Local configuration file used}}{}
756 </TS1sty>
```

60.1 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.

```
757 (*TS1oldsty)
758 \ProvidesPackage{textcomp}
759 [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-TeX world provide only this subset.

#4 = **#5** + `\texteuro`. Most newer fonts provide this.

#3 = **#4** + `\textomega`. Can also be described as $TS1 \cap (ISO-Adobe \cup MacRoman)$. (Except for the missing "currency".)

#2 = **#3** + `\textestimated` + `\textcurrency`. Can also be described as $TS1 \cap Adobe-Western-2$. This may be relevant for OpenType fonts, which usually show the Adobe-Western-2 character set.

#1 = TS1 without `\textcircled` and `\t`. These two glyphs are often not implemented and if their kernel defaults are changed commands like `\copyright` unnecessarily fail.

#0 = full TS1

And here a summary to go in the transcript file:

```

760 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
761   \space\space 5 = only ISO-Adobe without
762                       \string\textcurrency\MessageBreak
763   \space\space 4 = 5 + \string\texteuro\MessageBreak
764   \space\space 3 = 4 + \string\textohm\MessageBreak
765   \space\space 2 = 3 + \noexpand\textestimated+
766                       \string\textcurrency\MessageBreak
767   \space\space 1 = TS1 - \noexpand\textcircled-
768                       \string\t\MessageBreak
769   \space\space 0 = TS1 (full)\MessageBreak
770   Font families with sub-encoding setting implement\MessageBreak
771   only a restricted character set as indicated.\MessageBreak
772   Family '?' is the default used for unknown fonts.\MessageBreak
773   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 ?.

```

774 \def\DeclareEncodingSubset#1#2#3{%
775   \@ifundefined{#1:#2}%
776     {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
777     {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
778   \@namedef{#1:#2}{#3}}
779 \@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

```
780 \newif\iftc@forced \tc@forcedfalse
```

This is implemented by defining the default subset:

```
781 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
782 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
783 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
784 \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.

```
785 \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.

```
786 \def\tc@errorwarn{\PackageError}
787 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
788 \DeclareOption{quiet}{\gdef\tc@errorwarn#1#2#3{}}
789 \ExecuteOptions{almostfull}
790 \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.

```
791 \iftc@forced
```

If the “force” option was given we always use the default for testing against.

```

792 \def\CheckEncodingSubset#1#2#3#4#5{%
793   \ifnum #4>%
794     0\csname #2:\endcsname
795     \relax
796   \expandafter\@firstoftwo
797   \else
798   \expandafter\@secondoftwo
799   \fi
800   {#1{#2}}{#3}%
801   #5%
802 }

```

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.

```

803 \else
804 \def\CheckEncodingSubset#1#2#3#4#5{%
805   \ifnum #4>%
806     \expandafter\ifx\csname #2:\f@family\endcsname\relax
807     0\csname #2:\endcsname
808     \else
809     \csname #2:\f@family\endcsname
810     \fi
811     \relax
812   \expandafter\@firstoftwo
813   \else
814   \expandafter\@secondoftwo
815   \fi
816   {#1{#2}}{#3}%
817   #5%
818 }
819 \fi

```

$\backslash tc@subst$

```

820 \def\tc@subst#1{%
821   \tc@errorwarn{textcomp}%
822   {Symbol \string#1 not provided by\MessageBreak
823   font family \f@family\space
824   in TS1 encoding.\MessageBreak Default family used instead}\@eha
825   \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
826 }

```

$\backslash tc@error$ $\backslash tc@error$ 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.

```

827 % error commands take argument:
828 % #1 symbol to be used
829 \def\tc@error#1{%
830   \PackageError{textcomp}% % should be latex error if general
831   {Accent \string#1 not provided by\MessageBreak
832   font family \f@family\space
833   in TS1 encoding}\@eha
834 }

```

`\tc@fake@euro` `\tc@fake@euro` is an example of a “fake” definition to use in arg #3 of `\CheckEncodingSubset` when a symbol is not available in a certain font family. Here we produce an Euro symbol by combining a “C” with a “=”.

```

835 \def\tc@fake@euro#1{%
836   \leavevmode
837   \PackageInfo{textcomp}{Faking \noexpand#1for font family
838                 \f@family\MessageBreak in TS1 encoding}%
839   \valign{##\cr
840     \vfil\hbox to 0.07em{\dimen@\f@size\p@
841                       \math@fontsfalse
842                       \fontsize{.7\dimen@}\z@\selectfont=\hss}%
843     \vfil\cr%
844     \hbox{C}\crrr
845   }%
846 }

```

`\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.

```

847 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
848 \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...

```

849 \DeclareTextAccentDefault{\capitalcedilla}{TS1}
850 \DeclareTextAccentDefault{\capitalogonek}{TS1}
851 \DeclareTextAccentDefault{\capitalgrave}{TS1}
852 \DeclareTextAccentDefault{\capitalacute}{TS1}
853 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}
854 \DeclareTextAccentDefault{\capitaltilde}{TS1}
855 \DeclareTextAccentDefault{\capitaldieresis}{TS1}
856 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
857 \DeclareTextAccentDefault{\capitalring}{TS1}
858 \DeclareTextAccentDefault{\capitalcaron}{TS1}
859 \DeclareTextAccentDefault{\capitalbreve}{TS1}
860 \DeclareTextAccentDefault{\capitalmacron}{TS1}
861 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}

```

... and then the other glyphs.

```

862 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
863 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
864 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
865 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
866 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}
867 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
868 \DeclareTextSymbolDefault{\textdollar}{TS1}
869 \DeclareTextSymbolDefault{\textquotesingle}{TS1}
870 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
871 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}

```

```

872 \DeclareTextSymbolDefault{\textminus}{TS1}
873 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
874 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
875 \DeclareTextSymbolDefault{\textasciigrave}{TS1}
876 \DeclareTextSymbolDefault{\texttildelow}{TS1}
877 \DeclareTextSymbolDefault{\textasciibreve}{TS1}
878 \DeclareTextSymbolDefault{\textasciicaron}{TS1}
879 \DeclareTextSymbolDefault{\textgravedbl}{TS1}
880 \DeclareTextSymbolDefault{\textacutedbl}{TS1}
881 \DeclareTextSymbolDefault{\textdagger}{TS1}
882 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
883 \DeclareTextSymbolDefault{\textbardbl}{TS1}
884 \DeclareTextSymbolDefault{\textperthousand}{TS1}
885 \DeclareTextSymbolDefault{\textbullet}{TS1}
886 \DeclareTextSymbolDefault{\textcelsius}{TS1}
887 \DeclareTextSymbolDefault{\textflorin}{TS1}
888 \DeclareTextSymbolDefault{\texttrademark}{TS1}
889 \DeclareTextSymbolDefault{\textcent}{TS1}
890 \DeclareTextSymbolDefault{\textsterling}{TS1}
891 \DeclareTextSymbolDefault{\textyen}{TS1}
892 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
893 \DeclareTextSymbolDefault{\textsection}{TS1}
894 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}
895 \DeclareTextSymbolDefault{\textcopyright}{TS1}
896 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
897 \DeclareTextSymbolDefault{\textlnot}{TS1}
898 \DeclareTextSymbolDefault{\textregistered}{TS1}
899 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
900 \DeclareTextSymbolDefault{\textdegree}{TS1}
901 \DeclareTextSymbolDefault{\textpm}{TS1}
902 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
903 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
904 \DeclareTextSymbolDefault{\textasciacute}{TS1}
905 \DeclareTextSymbolDefault{\textmu}{TS1}
906 \DeclareTextSymbolDefault{\textparagraph}{TS1}
907 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
908 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
909 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
910 \DeclareTextSymbolDefault{\textonequarter}{TS1}
911 \DeclareTextSymbolDefault{\textonehalf}{TS1}
912 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
913 \DeclareTextSymbolDefault{\texttimes}{TS1}
914 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

915 \DeclareTextCommandDefault{\texteuro}
916 {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```

917 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol14\textohm}

```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```

918 \DeclareTextCommandDefault{\textestimated}%
919   {\tc@check@symbol3\textestimated}
920 \DeclareTextCommandDefault{\textcurrency}%
921   {\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.

```

922 \DeclareTextCommandDefault{\capitaltie}%
923   {\tc@check@accent2\capitaltie}
924 \DeclareTextCommandDefault{\newtie}%
925   {\tc@check@accent2\newtie}
926 \DeclareTextCommandDefault{\capitalnewtie}%
927   {\tc@check@accent2\capitalnewtie}
928 \DeclareTextCommandDefault{\textleftarrow}%
929   {\tc@check@symbol2\textleftarrow}
930 \DeclareTextCommandDefault{\textrightarrow}%
931   {\tc@check@symbol2\textrightarrow}
932 \DeclareTextCommandDefault{\textblank}%
933   {\tc@check@symbol2\textblank}
934 \DeclareTextCommandDefault{\textdblhyphen}%
935   {\tc@check@symbol2\textdblhyphen}
936 \DeclareTextCommandDefault{\textzerooldstyle}%
937   {\tc@check@symbol2\textzerooldstyle}
938 \DeclareTextCommandDefault{\textoneoldstyle}%
939   {\tc@check@symbol2\textoneoldstyle}
940 \DeclareTextCommandDefault{\texttwooldstyle}%
941   {\tc@check@symbol2\texttwooldstyle}
942 \DeclareTextCommandDefault{\textthreeoldstyle}%
943   {\tc@check@symbol2\textthreeoldstyle}
944 \DeclareTextCommandDefault{\textfouroldstyle}%
945   {\tc@check@symbol2\textfouroldstyle}
946 \DeclareTextCommandDefault{\textfiveoldstyle}%
947   {\tc@check@symbol2\textfiveoldstyle}
948 \DeclareTextCommandDefault{\textsixoldstyle}%
949   {\tc@check@symbol2\textsixoldstyle}
950 \DeclareTextCommandDefault{\textsevenoldstyle}%
951   {\tc@check@symbol2\textsevenoldstyle}
952 \DeclareTextCommandDefault{\texteightoldstyle}%
953   {\tc@check@symbol2\texteightoldstyle}
954 \DeclareTextCommandDefault{\textnineoldstyle}%
955   {\tc@check@symbol2\textnineoldstyle}
956 \DeclareTextCommandDefault{\textlangle}%
957   {\tc@check@symbol2\textlangle}
958 \DeclareTextCommandDefault{\textrightangle}%
959   {\tc@check@symbol2\textrightangle}
960 \DeclareTextCommandDefault{\textmho}%
961   {\tc@check@symbol2\textmho}
962 \DeclareTextCommandDefault{\textbigcircle}%
963   {\tc@check@symbol2\textbigcircle}
964 \DeclareTextCommandDefault{\textuparrow}%
965   {\tc@check@symbol2\textuparrow}
966 \DeclareTextCommandDefault{\textdownarrow}%
967   {\tc@check@symbol2\textdownarrow}
968 \DeclareTextCommandDefault{\textborn}%

```

```

969     {\tc@check@symbol2\textborn}
970 \DeclareTextCommandDefault{\textdivorced}%
971     {\tc@check@symbol2\textdivorced}
972 \DeclareTextCommandDefault{\textdied}%
973     {\tc@check@symbol2\textdied}
974 \DeclareTextCommandDefault{\textleaf}%
975     {\tc@check@symbol2\textleaf}
976 \DeclareTextCommandDefault{\textmarried}%
977     {\tc@check@symbol2\textmarried}
978 \DeclareTextCommandDefault{\textmusicalnote}%
979     {\tc@check@symbol2\textmusicalnote}
980 \DeclareTextCommandDefault{\textdblhyphenchar}%
981     {\tc@check@symbol2\textdblhyphenchar}
982 \DeclareTextCommandDefault{\textdollaroldstyle}%
983     {\tc@check@symbol2\textdollaroldstyle}
984 \DeclareTextCommandDefault{\textcentoldstyle}%
985     {\tc@check@symbol2\textcentoldstyle}
986 \DeclareTextCommandDefault{\textcolonmonetary}%
987     {\tc@check@symbol2\textcolonmonetary}
988 \DeclareTextCommandDefault{\textwon}%
989     {\tc@check@symbol2\textwon}
990 \DeclareTextCommandDefault{\textnaira}%
991     {\tc@check@symbol2\textnaira}
992 \DeclareTextCommandDefault{\textguarani}%
993     {\tc@check@symbol2\textguarani}
994 \DeclareTextCommandDefault{\textpeso}%
995     {\tc@check@symbol2\textpeso}
996 \DeclareTextCommandDefault{\textlira}%
997     {\tc@check@symbol2\textlira}
998 \DeclareTextCommandDefault{\textrecipe}%
999     {\tc@check@symbol2\textrecipe}
1000 \DeclareTextCommandDefault{\textinterrobang}%
1001     {\tc@check@symbol2\textinterrobang}
1002 \DeclareTextCommandDefault{\textinterrobangdown}%
1003     {\tc@check@symbol2\textinterrobangdown}
1004 \DeclareTextCommandDefault{\textdong}%
1005     {\tc@check@symbol2\textdong}
1006 \DeclareTextCommandDefault{\textpertenthousand}%
1007     {\tc@check@symbol2\textpertenthousand}
1008 \DeclareTextCommandDefault{\textpilcrow}%
1009     {\tc@check@symbol2\textpilcrow}
1010 \DeclareTextCommandDefault{\textbaht}%
1011     {\tc@check@symbol2\textbaht}
1012 \DeclareTextCommandDefault{\textnumero}%
1013     {\tc@check@symbol2\textnumero}
1014 \DeclareTextCommandDefault{\textdiscount}%
1015     {\tc@check@symbol2\textdiscount}
1016 \DeclareTextCommandDefault{\textopenbullet}%
1017     {\tc@check@symbol2\textopenbullet}
1018 \DeclareTextCommandDefault{\textservicemark}%
1019     {\tc@check@symbol2\textservicemark}
1020 \DeclareTextCommandDefault{\textlquill}%
1021     {\tc@check@symbol2\textlquill}
1022 \DeclareTextCommandDefault{\textrquill}%

```

```

1023   {\tc@check@symbol2\texttrquill}
1024 \DeclareTextCommandDefault{\textcopyleft}%
1025   {\tc@check@symbol2\textcopyleft}
1026 \DeclareTextCommandDefault{\textcircledP}%
1027   {\tc@check@symbol2\textcircledP}
1028 \DeclareTextCommandDefault{\textreferencemark}%
1029   {\tc@check@symbol2\textreferencemark}
1030 \DeclareTextCommandDefault{\textsurd}%
1031   {\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 math font encodings, i.e., the third argument to `\CheckEncodingSubset` uses `\UseTextAccent` to get them from there.

```

1032 \DeclareTextCommandDefault{\textcircled}
1033   {\CheckEncodingSubset\UseTextAccent{TS1}%
1034     {\UseTextAccent{OMS}}1\textcircled}
1035 \DeclareTextCommandDefault{\t}
1036   {\CheckEncodingSubset\UseTextAccent{TS1}%
1037     {\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).

```
1038 \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:

```

1039 \UndeclareTextCommand{\textsterling}{OT1}
1040 \UndeclareTextCommand{\textdollar} {OT1}

```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```

1041 %\UndeclareTextCommand{\textsterling}{OT4}
1042 %\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`.

```

1043 \UndeclareTextCommand{\textperthousand}{T1}
1044 %\UndeclareTextCommand{\textpertenthousand}{T1}

```

60.1.1 Supporting oldstyle digits

```
1045 \DeclareRobustCommand\oldstylenums[1]{%
1046 \begingroup
1047 \ifmmode
1048 \mathgroup\symletters #1%
1049 \else
1050 \CheckEncodingSubset\@use@text@encoding{TS1}%
1051 {\PackageWarning{textcomp}%
1052 {Oldstyle digits unavailable for
1053 family \f@family.\MessageBreak
1054 Lining digits used instead}}%
1055 \tw@{#1}%
1056 \fi
1057 \endgroup
1058 }
```

60.1.2 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:

```
1059 \iftc@forced \else
```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```
1060 \DeclareEncodingSubset{TS1}{cmr} {0}
1061 \DeclareEncodingSubset{TS1}{cmss} {0}
1062 \DeclareEncodingSubset{TS1}{cmtt} {0}
1063 \DeclareEncodingSubset{TS1}{cmvtt} {0}
1064 \DeclareEncodingSubset{TS1}{cmbr} {0}
1065 \DeclareEncodingSubset{TS1}{cmtl} {0}
1066 \DeclareEncodingSubset{TS1}{ccr} {0}
```

PSNFSS fonts:

```
1067 \DeclareEncodingSubset{TS1}{ptm} {4}
1068 \DeclareEncodingSubset{TS1}{pcr} {4}
1069 \DeclareEncodingSubset{TS1}{phv} {4}
1070 \DeclareEncodingSubset{TS1}{ppl} {3}
1071 \DeclareEncodingSubset{TS1}{pag} {4}
1072 \DeclareEncodingSubset{TS1}{pbk} {4}
1073 \DeclareEncodingSubset{TS1}{pnc} {4}
1074 \DeclareEncodingSubset{TS1}{pzc} {4}
1075 \DeclareEncodingSubset{TS1}{bch} {4}
1076 \DeclareEncodingSubset{TS1}{put} {5}
```

Other CTAN fonts (probably not complete):

```
1077 \DeclareEncodingSubset{TS1}{uag} {5}
1078 \DeclareEncodingSubset{TS1}{ugq} {5}
1079 \DeclareEncodingSubset{TS1}{ul8} {4}
1080 \DeclareEncodingSubset{TS1}{ul9} {4} % (LuxiSans, one day)
1081 \DeclareEncodingSubset{TS1}{augie} {5}
1082 \DeclareEncodingSubset{TS1}{dayrom} {3}
1083 \DeclareEncodingSubset{TS1}{dayroms} {3}
1084 \DeclareEncodingSubset{TS1}{pxr} {0}
```

```

1085 \DeclareEncodingSubset{TS1}{pxss} {0}
1086 \DeclareEncodingSubset{TS1}{pxtt} {0}
1087 \DeclareEncodingSubset{TS1}{txr} {0}
1088 \DeclareEncodingSubset{TS1}{txss} {0}
1089 \DeclareEncodingSubset{TS1}{txtt} {0}

```

Latin Modern and TeX Gyre:

```

1090 \DeclareEncodingSubset{TS1}{lmr} {0}
1091 \DeclareEncodingSubset{TS1}{lmdh} {0}
1092 \DeclareEncodingSubset{TS1}{lmss} {0}
1093 \DeclareEncodingSubset{TS1}{lmssq} {0}
1094 \DeclareEncodingSubset{TS1}{lmvtt} {0}
1095 \DeclareEncodingSubset{TS1}{lmtt} {0}

1096 \DeclareEncodingSubset{TS1}{qhv} {0}
1097 \DeclareEncodingSubset{TS1}{qag} {0}
1098 \DeclareEncodingSubset{TS1}{qbk} {0}
1099 \DeclareEncodingSubset{TS1}{qcr} {0}
1100 \DeclareEncodingSubset{TS1}{qcs} {0}
1101 \DeclareEncodingSubset{TS1}{qpl} {0}
1102 \DeclareEncodingSubset{TS1}{qtm} {0}
1103 \DeclareEncodingSubset{TS1}{qzc} {0}
1104 \DeclareEncodingSubset{TS1}{qhvc} {0}

```

Fourier-GUTenberg:

```

1105 \DeclareEncodingSubset{TS1}{futs} {4}
1106 \DeclareEncodingSubset{TS1}{futex} {4}
1107 \DeclareEncodingSubset{TS1}{futj} {4}

```

Y&Y's Lucida Bright

```

1108 \DeclareEncodingSubset{TS1}{hlh} {3}
1109 \DeclareEncodingSubset{TS1}{hls} {3}
1110 \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`.

```

1111 \DeclareEncodingSubset{TS1}{hlct} {5}
1112 \DeclareEncodingSubset{TS1}{hlx} {5}
1113 \DeclareEncodingSubset{TS1}{hlce} {5}
1114 \DeclareEncodingSubset{TS1}{hlcn} {5}
1115 \DeclareEncodingSubset{TS1}{hlcw} {5}
1116 \DeclareEncodingSubset{TS1}{hlcf} {5}

```

Other commercial families...

```

1117 \DeclareEncodingSubset{TS1}{pplx} {3}
1118 \DeclareEncodingSubset{TS1}{pplj} {3}
1119 \DeclareEncodingSubset{TS1}{ptmx} {4}
1120 \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.

```

1121 \InputIfFileExists{textcomp.cfg}
1122 {\PackageInfo{textcomp}{Local configuration file used}}{}

```

1123 \fi

1124 </TS1oldsty>

File y

ltpageno.dtx

61 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

62 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 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.

62.1 Cross Referencing

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

```
1 (*2kernel)
2 \message{x-ref,}
```

This is implemented as follows. A referencable counter CNT is incremented by the command `\refstepcounter{CNT}`, which sets `\@currentlabel == {CNT}{eval(\p@cnt\theCNT)}`. The command `\label{FOO}` then writes the following on file `\@auxout`:

```
\newlabel{FOO}{\eval{\@currentlabel}}{\eval{\thepage}}
```

```
\ref{FOO} ==
BEGIN
  if \r@foo undefined
  then @refundefined := G T
      ??
      Warning: 'reference foo on page ... undefined'
  else \@car \eval{\r@FOO}\@nil
  fi
END
```

```
\pageref{foo} =
BEGIN
  if \r@foo undefined
  then @refundefined := G T
      ??
      Warning: 'reference foo on page ... undefined'
  else \@cdr \eval{\r@FOO}\@nil
  fi
END
```

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 </2kernel>
37 < *2kernel | 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`.¹⁰

```
45 \DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%  
46   \expandafter\MakeUppercase\@tempa}
```

```
47 </2kernel | latexrelease>  
48 (latexrelease)\EndIncludeInRelease  
49 (latexrelease)\IncludeInRelease{0000/00/00}%  
50 (latexrelease)           {\refstepcounter}{Add \labelformat and \Ref}%  
51 (latexrelease)  
52 (latexrelease)\def\refstepcounter#1{\stepcounter{#1}%  
53 (latexrelease)   \protected@edef\@currentlabel  
54 (latexrelease)     {\csname p@#1\endcsname\csname the#1\endcsname}%  
55 (latexrelease)}  
56 (latexrelease)\let\labelformat\undefined  
57 (latexrelease)\let\Ref\undefined  
58 (latexrelease)  
59 (latexrelease)\EndIncludeInRelease  
60 <*2kernel)
```

`\@currentlabel` Default for `\label` commands that come before any environment.

```
61 \def\@currentlabel{}  
  
62 </2kernel)
```

¹⁰There 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

63 Miscellaneous Environments

This section implements the basic environment mechanism, and also a few specific environments including `document`, The math environments and related commands, the ‘flushing’ environments, (`center`, `flushleft`, `flushright`), and `verbatim`.

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

63.1 Environments

`\begin{foo}` and `\end{foo}` are used to delimit environment `foo`.

`\begin{foo}` starts a group and calls `\foo` if it is defined, otherwise it does nothing.

`\end{foo}` checks to see that it matches the corresponding `\begin` and if so, it calls `\endfoo` and does an `\endgroup`. Otherwise, `\end{foo}` does nothing.

If `\end{foo}` needs to ignore blanks after it, then `\endfoo` should globally set the `@ignore` switch true with `\@ignoretrue` (this will automatically be global).

NOTE: `\@@end` is defined to be the `\end` command of T_EX82.

`\enddocument` is the user’s command for ending the manuscript file.

`\stop` is a panic button — to end T_EX in the middle.

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

```
\enddocument ==
BEGIN
  \@checkend{document}    %% checks for unmatched \begin
  \clearpage
  \beginingroup
  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 @tempswa := true      fi
  END
  \bibcite{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     \@tempwafalse
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 \ifx\protected@file@percent\@percent
44 \else
45 \ifx\protected@file@percent\@percent
46 \else
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 </2ekernel | latexrelease>
71 <latexrelease>\EndIncludeInRelease
72 <latexrelease>\IncludeInRelease{0000/00/00}%
73 <latexrelease>           {\protected@file@percent}{Mask line endings}%
74 <latexrelease>\let\protected@file@percent\@undefined
75 <latexrelease>\let\add@percent@to@temptokena\@undefined
76 <latexrelease>\let\do@add@percent@to@temptokena\@undefined
77 <latexrelease>\let\dont@add@percent@to@temptokena\@undefined
78 <latexrelease>\long\def\@writefile#1#2{%
79 <latexrelease>  \@ifundefined{tf@#1}\relax
80 <latexrelease>    {\@temptokena{#2}}%
81 <latexrelease>    \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
82 <latexrelease>    }%
83 <latexrelease>}
84 <latexrelease>\EndIncludeInRelease
85 <*/2ekernel>

```

`\stop`

```

86 \def\stop{\clearpage\deadcycles\z@ \let\par\@par\@end}

```

Historical L^AT_EX 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

```

An 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

```

63.2 Center, Flushright, Flushleft

```
152 \message{center,}
```

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

```
\center, \flushright and \flushleft set
  \rightskip = Opt or \@flushglue (as appropriate)
  \leftskip  = Opt or \@flushglue (as appropriate)
  \parindent = Opt
  \parfillskip = Opt. (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 </2ekernel>
154 (*2ekernel | 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 </2ekernel | 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 (*2ekernel)
```

```
\@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 \</2kernel>
175 \*2kernel | latexrelease>
176 \<latexrelease>\IncludeInRelease{2019/10/01}%
177 \<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 \</2kernel | latexrelease>
191 \<latexrelease>\EndIncludeInRelease
192 \<latexrelease>\IncludeInRelease{0000/00/00}%
193 \<latexrelease> \centering}{Make commands robust}%
194 \<latexrelease>
195 \<latexrelease>\kernel@make@fragile\centering
196 \<latexrelease>\kernel@make@fragile\raggedright
197 \<latexrelease>\kernel@make@fragile\raggedleft
198 \<latexrelease>
199 \<latexrelease>\EndIncludeInRelease
200 \*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}

```

63.3 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 \TeX book'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 \@tempwafalse
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) \@tempwafalse
246 (latexrelease) \def\par{%
247 (latexrelease) \if@tempwa
248 (latexrelease) \leavevmode \null \@@par\penalty\interlinepenalty
249 (latexrelease) \else
250 (latexrelease) \@tempwatrue
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 \@latex@error{\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 </2ekernel | 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 (*2ekernel)

```

\@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 </2ekernel)
```

File B

l^AT_EX math.dtx

64 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,}
```

64.1 Math commands based on plain T_EX

64.1.1 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)}

```

64.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```

41 \DeclareRobustCommand\bigl{\mathopen\big}
42 \DeclareRobustCommand\bigm{\mathrel\big}
43 \DeclareRobustCommand\bigg{\mathclose\big}
44 \DeclareRobustCommand\Bigl{\mathopen\Big}
45 \DeclareRobustCommand\Bigm{\mathrel\Big}
46 \DeclareRobustCommand\Bigg{\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}

```

64.1.3 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@t 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 (/2kernel)
78 (*2kernel | 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 (/2kernel | 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 (*2kernel)

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 </2kernel>
108 <*2kernel | 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 </2kernel | 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 <*2kernel>

```

\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 </2kernel>
138 <*2kernel | 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 </2kernel | 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 <*2kernel>

```

\buildrel

```

149 \def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}

150 </2kernel>
151 <*2kernel | 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{##$\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
177       &\quad\hfil$##$\hfil\crr
178       \omit\strut\hfil\crr\noalign{\kern-\baselineskip}%
179       #1\crr\omit\strut\cr}}%
180   \setbox\tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
181   \setbox\hw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
182   \setbox\tw@\hbox{\$ \kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
183     \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
184     \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,\right)$}%
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\disply{\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{${@align\hfil\displaystyle##\hfil$}\cr
198   #1\cr}}

\sp
\sb 199 \let\sp=^
    200 \let\sb=_

\>
\; 201 %\def\,{\mskip\thinmuskip} % already defined in ltspace
\! 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 \_

```

64.2 Math Environments

`\(` Produces \dots with checks that `\(` isn't used in math mode, and that `\)` is only used in math mode begun with `\(`.

```
225 </2ekernel>
226 (latexrelease)\IncludeInRelease{2015/01/01}{\({Make \(\ robust}%
227 (*2ekernel | latexrelease)
228 \DeclareRobustCommand\({%
229   \relax\ifmmode\@badmath\else$\fi}%
230 \DeclareRobustCommand\)%{%
231   \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
232 </2ekernel | latexrelease)
233 (latexrelease)\EndIncludeInRelease
234 (latexrelease)\IncludeInRelease{0000/00/00}{\({Make \(\ robust}%
235 (latexrelease)\def\({%
236 (latexrelease)   \relax\ifmmode\@badmath\else$\fi}%
237 (latexrelease)\expandafter\let\csname\string( \endcsname\@undefined
238 (latexrelease)\def\)%{%
239 (latexrelease)   \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
240 (latexrelease)\expandafter\let\csname\string) \endcsname\@undefined
241 (latexrelease)\EndIncludeInRelease
242 (*2ekernel)
```

`\[` 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 </2ekernel>
244 (latexrelease)\IncludeInRelease{2015/01/01}{\[Make \[ robust}%
245 (*2ekernel | latexrelease)
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 </2ekernel | latexrelease)
```

```

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 $\langle \dots \rangle$ and $\llbracket \dots \rrbracket$.

```

displaymath 299 \let\math=\langle
300 \let\endmath=\rangle
301 \def\displaymath{\llbracket}
302 \def\enddisplaymath{\rrbracket\@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{\font\@normalfont \normalcolor (\theequation)}

```

\stackrel A disguise for plain T_EX's `\stackrel`.

```

307 \DeclareRobustCommand\stackrel[2]{\mathrel{\mathop{\#2}\limits^{\#1}}}

```

```

\frac A disguise for plain TEX's \over.
308 \DeclareRobustCommand\frac[2]{\begingroup#1\endgroup\over#2}}

\sqrt Add an optional argument to plain's \sqrt to give the nth 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\@eqnset = \hfil.
\@eqpen 311 \newcount\@eqcnt
\if@eqnsw 312 \newcount\@eqpen
\@eqnset 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\\\@eqnscr
324   $$\everycr{}\halign to\displaywidth\bgroup
325     \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnset
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@@ \hb@xt@\z@\bgroup\hss##\egroup
330     \tabskip\z@skip
331   \cr
332 }

333 \def\endeqnarray{%
334   \@eqnscr
335   \egroup
336   \global\advance\c@equation\m@ne
337   $$\@ignoretrue
338 }

339 \let\@eqnset=\relax

\nonumber Switches off equation numbering.
340 \def\nonumber{\global\@eqnswfalse}

\@eqnscr
\@xeqnscr 341 \def\@eqnscr{%
\@yeqnscr 342   {\ifnum0='}\fi
343   \@ifstar{%
344     \global\@eqpen\M\@yeqnscr

```

```

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 \latex@error{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 </2kernel)

```

64.3 External options to the standard document classes

64.3.1 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)

```

64.3.2 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}

\l[ Begin display math;
382 \IncludeInRelease{2015/01/01}{\l[}{Make \l[ robust}%
383 \DeclareRobustCommand\l[{\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}{\l[}{Make \l[ robust}%
395 \renewcommand\l[{\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

\l] end display math;
406 \IncludeInRelease{2015/01/01}{\l]}{Make \l] robust}%
407 \DeclareRobustCommand\l]{\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             \ifmode
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\@=\@eqnrcr
443   \setlength\abovedisplayskip{\topsep}%
444   \ifmode
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   $\displaystyle\tabskip\z@skip{##}$\@eqnsele%
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

65 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.

65.1 List and Trivlist

The following variables are used inside a list environment:

`\totalleftmargin` The distance that the prevailing left margin is indented from the outermost left margin,

`\linewidth` The width of the current line. Must be initialized to `\hsize`.

`\listdepth` A count for holding current list nesting depth.

`\makelabel` A macro with a single argument, used to generate the label from the argument (given or implied) of the `\item` command. Initialized to `\@mklab` by the `\list` command. This command must produce some stretch—i.e., an `\hfil`.

`\@inlabel` A switch that is false except between the time an `\item` is encountered and the time that `TEX` actually enters horizontal mode. Should be tested by commands that can be messed up by the list environment's use of `\everypar`.

`\box\@labels` When `@inlabel = true`, it holds the labels to be put out by `\everypar`.

`@noparitem` A switch set by `\list` when `@inlabel = true`. Handles the case of a `\list` being the first thing in an item.

`@noparlist` A switch set true for a list that begins an item. No `\topsep` space is added before or after `\item`'s such a list.

`@newlist` Set true by `\list`, set false by the first text (by `\everypar`).

`@noitemarg` Set true when executing an `\item` with no explicit argument. Used to save space. To save time, make two separate `\@item` commands.

`@nmbri` 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.

65.2 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.

65.3 Penalties

`\@beginparpenalty`: put at the beginning of a list

`\@endparpenalty`: put at end of list

`\@itempenalty`: put between items.

65.4 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.

65.5 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
  @nمبرlist := 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 \@tempkipa := \lastskip
           \vskip - \lastskip
           \vskip \@tempkipa - \@outerparskip + \parskip
    fi
    \@endparenv
  fi
END

\@endparenv ==
BEGIN
  \addpenalty{@endparpenalty}
  \addvspace{\@topsepadd}

```

```

\endgroup    %% ends the \begin command's \begingroup
\par == BEGIN
            \@restorepar
            \everypar{}
            \par
            END
\everypar == BEGIN remove \lastbox \everypar{} END
\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
      \@tempkipa := \lastskip
      \vskip -\lastskip
      \vskip \@tempkipa + \@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 @nobreake = 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{} }
@nobraek :=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@noperitem
\@noperitemfalse 30 \newif\if@noperitem \@noperitemfalse
\@noperitemtrue
\if@noperlist
\@noperlistfalse 31 \newif\if@noperlist \@noperlistfalse
\@noperlisttrue
\if@noitemarg
\@noitemargfalse 32 \newif\if@noitemarg \@noitemargfalse
\@noitemargtrue
\if@newlist
\@newlistfalse 33 \newif\if@nmbriest \@nmbriestfalse
\@newlisttrue
\list
34 \def\list#1#2{%
35 \ifnum \@listdepth >5\relax
36 \@toodeep
37 \else
38 \global\advance\@listdepth\@ne

```

```

39 \fi
40 \rightmargin\z@
41 \listparindent\z@
42 \itemindent\z@
43 \csname @list\romannumeral\the\@listdepth\endcsname
44 \def\@itemlabel{#1}%
45 \let\makelabel\@mklab
46 \@nmbrrlistfalse
47 #2\relax
48 \@trivlist
49 \parskip\parsep
50 \parindent\listparindent
51 \advance\linewidth -\rightmargin
52 \advance\linewidth -\leftmargin
53 \advance\@totalleftmargin \leftmargin
54 \parshape \@ne \@totalleftmargin \linewidth
55 \ignorespaces}

```

\par@deathcycles

```
56 \newcount\par@deathcycles
```

\@trivlist Because \par is sometimes made a no-op it is possible for a missing \item to produce a loop that does not fill memory and so never gets trapped by T_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     \@nmbulistfalse
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 `\@currenenvir` resolves suitably). Otherwise the usual "perhaps a missing item" error will get triggered later which is confusing.

```

111     \else
112         \@inmatherr{\end{\@currenenvir}}}%
113     \fi
114     \if@noparlist \else
115         \ifdim\lastskip >\z@
116             \@tempskipa\lastskip \vskip -\lastskip
117             \advance\@tempskipa\parskip \advance\@tempskipa -\@outerparskip
118             \vskip\@tempskipa
119         \fi
120         \@endparenv
121     \fi
122 }

```

`\@endparenv` To suppress the paragraph indentation in text immediately following a paragraph-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.

```

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 `\@nobreak` when it is true is now essential since now it is sometimes set locally.

```

192   \if@nobreak
193     \@nobreakfalse
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@}

```

65.6 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@enumiii 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 </2kernel)

```

File D

ltboxes.dtx

66 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[⟨raised⟩]{⟨width⟩}{⟨height⟩} : Makes a ⟨width⟩ * ⟨height⟩ rule, raised
⟨raised⟩.
\underline \underline{⟨text⟩} : Makes an underlined hbox with ⟨text⟩ in it.
\raisebox \raisebox{⟨distance⟩}[⟨height⟩][⟨depth⟩]{⟨box⟩} :
Raises ⟨box⟩ up by ⟨distance⟩ length (down if ⟨distance⟩ negative). Makes TEX
think that the new box extends ⟨height⟩ above the line and ⟨depth⟩ below, for a
total vertical length of ⟨height⟩+⟨depth⟩. Default values of ⟨height⟩ & ⟨depth⟩ =
actual height and depth of box in new position.
1 ⟨*2kernel⟩
2 \message{boxes,}

\makebox \makebox User level command just looks for optional [ or (.
3 ⟨/2kernel⟩
4 ⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
5 ⟨latexrelease⟩           {\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 L^AT_EX.

```
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
`\color@endgroup` be `\begingroup`, `\endgroup`, `\begingroup\set@color`,
`\color@setgroup` `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and *(set to main*
`\normalcolor` *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 `/2kernel)`
72 `(latexrelease)\IncludeInRelease{2015/01/01}%`
73 `(latexrelease) \savebox}{Make \savebox robust}%`
74 `(*2kernel | latexrelease)`
75 `\DeclareRobustCommand\savebox[1]{%`
76 `\@ifnextchar(%)`
77 `{\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%`
78 `/2kernel | 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\cename savebox \endcename\undefined
86 (latexrelease)\EndIncludeInRelease
87 (*2ekernel)

\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 \endpefalse
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 \@framebox\relax}

\framebox Framed version of \makebox.
135 </2ekernel>
136 <latexrelease>\IncludeInRelease{2015/01/01}%
137 <latexrelease> \framebox}{Make \framebox robust}%
138 <*2ekernel | latexrelease>
139 \DeclareRobustCommand\framebox{%
140 \@ifnextchar(%)
141 \@framepicbox{\@ifnextchar[\@framebox\fbox]}%
142 </2ekernel | latexrelease>
143 <latexrelease>\EndIncludeInRelease
144 <latexrelease>\IncludeInRelease{0000/00/00}%
145 <latexrelease> \framebox}{Make \framebox robust}%
146 <latexrelease>\def\framebox{%
147 <latexrelease> \@ifnextchar(%)
148 <latexrelease> \@framepicbox{\@ifnextchar[\@framebox\fbox]}%
149 <latexrelease>\expandafter\let\csname framebox \endcsname\@undefined
150 <latexrelease>\EndIncludeInRelease
151 <*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\fbboxsep\csname bm@#2\endcsname\kern\fbboxsep}%
162   \@frameb@x{\kern-\fbboxrule}%
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\fbboxrule
166   \advance\@tempdima\fbboxsep
167   \advance\@tempdima\dp\@tempboxa
168   \hbox{%
169     \lower\@tempdima\hbox{%
170       \vbox{%
171         \hrule\@height\fbboxrule
172         \hbox{%
173           \vrule\@width\fbboxrule
174           #1%
175           \vbox{%
176             \vskip\fbboxsep
177             \box\@tempboxa
178             \vskip\fbboxsep}%
179           #1%
180           \vrule\@width\fbboxrule}%
181         \hrule\@height\fbboxrule}%
182         }%
183       }%
184   }

\@framepicbox Picture mode version.
185 \def\@framepicbox(#1,#2){%
186   \@ifnextchar[{\@iframepicbox(#1,#2)}{\@framepicbox(#1,#2) []}]

\@iframepicbox Picture mode version.
187 \long\def\@iframepicbox(#1,#2)[#3]#4{%
188   \frame{\@imakepicbox(#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 {\@iparbox{#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\ifmode\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@nobraek`; otherwise this command will be redundant.

```

235 </2ekernel>
236 <latexrelease>\IncludeInRelease{2017-04-15}%
237 <latexrelease>          {\normallineskiplimit}
238 <latexrelease>          {reset \lineskiplimit}%
239 <*2ekernel | latexrelease>
240 \def\@arrayparboxrestore{%
241   \let\if@nobraek\iffalse
242   \let\if@noskipsec\iffalse
243   \let\par\@@par
244   \let\-\@dischyph

```

Redefined accents to allow changes in font encoding

```

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 </2ekernel | latexrelease>

```

```

257 <latexrelease>\EndIncludeInRelease
258 <latexrelease>\IncludeInRelease{0000-00-00}%
259 <latexrelease>          {\normallineskiplimit}
260 <latexrelease>          {reset \lineskiplimit}%
261 <latexrelease>\def\@arrayparboxrestore{%
262 <latexrelease>  \let\if@nobraek\iffalse
263 <latexrelease>  \let\if@noskipsec\iffalse
264 <latexrelease>  \let\par\@@par
265 <latexrelease>  \let\-\@dischyph
266 <latexrelease>  \let'\@acci\let'\@accii\let\=\@acciii
267 <latexrelease>  \parindent\z@ \parskip\z@skip
268 <latexrelease>  \everypar{}%
269 <latexrelease>  \linewidth\hsize
270 <latexrelease>  \@totalleftmargin\z@
271 <latexrelease>  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
272 <latexrelease>  \parfillskip\@flushglue \lineskip\normallineskip
273 <latexrelease>  \baselineskip\normalbaselineskip
274 <latexrelease>  \sloppy}
275 <latexrelease>\EndIncludeInRelease
276 <*2ekernel>

```

```

\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     {\@iiminipage c\relax[s]}}

\@iminipage Optional argument handling.
285 \def\@iminipage[#1]{%
286   \ifnextchar[%
287     {\@iiminipage{#1}}%
288     {\@iiminipage{#1}\relax[s]}}

\@iiminipage Optional argument handling.
289 \def\@iiminipage#1[#2]{%
290   \ifnextchar[%
291     {\@iiminipage{#1}{#2}}%
292     {\@iiminipage{#1}{#2}[#1]}}

\@iiminipage Internal form of minipage.
293 \def\@iiminipage#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

```

```

\underscore LATEX version works outside math.
359 \DeclareRobustCommand\underscore[1]{%
360   \relax
361   \ifmode\@@underline{#1}%
362   \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}

\raisebox Raise a box, and change its vertical dimensions.
363 \ifkernel
364 (latexrelease)\IncludeInRelease{2015/01/01}%
365 (latexrelease)      {\raisebox}{Make \raisebox robust}%
366 (*2ekernel | latexrelease)
367 \DeclareRobustCommand\raisebox[1]{%
368   \leavevmode
369   \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1} []]}
370 \ifkernel | latexrelease)
371 (latexrelease)\EndIncludeInRelease
372 (latexrelease)\IncludeInRelease{0000/00/00}%
373 (latexrelease)      {\raisebox}{Make \raisebox robust}%
374 (latexrelease)\def\raisebox#1{%
375 (latexrelease)   \leavevmode
376 (latexrelease)   \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1} []]}
377 (latexrelease)\expandafter\let\cename raisebox \endcename\@undefined
378 (latexrelease)\EndIncludeInRelease
379 (*2ekernel)

\@rsbox Optional argument handling.
380 \def\@rsbox#1[#2]{%
381   \@ifnextchar[{\@iirsbox{#1}[#2]}{\@irsbox{#1}[#2]}}

\@argrsbox ...

\@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}
```

66.1 Some low-level constructs

The following commands are basically inherited from plain T_EX.

```
\leftline  These macros place text on a full line either centred or left or right adjusted.
\rightline 402 \def\@@line{\hb@xt@\hsize}
\centerline 403 \DeclareRobustCommand\leftline[1]{\@@line{#1\hss}}
\@@line     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

67 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.

67.1 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 \@nxttabmar < \@hightab
    then \@nxttabmar :=G \@nxttabmar+1
    else error message "Undefined tab"
    fi
END

\@tabminus ==
BEGIN
  if \@nxttabmar > \@firsttab
    then \@nxttabmar :=G \@nxttabmar-1
    else error message "Too many untab"
    fi
END

\@tabrj ==
BEGIN \@stopfield

```

```

        \@addfield
        @rjfield :=G T
        \@startfield
    END

\@tablab ==
BEGIN \@stopfield
    \box\@curline G:= \hbox{\box\@curline %% 'G' added 17 Jun 86
        \hskip - width of \box\@curfield
        \hskip -\tabbingsep
        \box\@curfield
        \hskip \tabbingsep }

    \@startfield
END

\pushtabs ==
BEGIN
    \@stopfield
    \@tabpush :=G \@tabpush + 1
    \begingroup
    \@contfield
END

\poptabs ==
BEGIN
    \@stopfield
    if \@tabpush > 0
        then \endgroup
            \@tabpush :=G \@tabpush - 1
        else error message: "Too many \poptabs"
    fi
    \@contfield
END

```

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

```

67.2 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

```

`\doublerulesep` : 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

\beginarray ==
BEGIN
  \@halignto == NULL
  \@tabular
END

\beginarray*{WIDTH} ==
BEGIN
  \@halignto == to WIDTH
  \@tabular
END

\beginarray ==
BEGIN
  \leavevmode
  \hbox { $
    \@acol == \@tabacol
    \@classz == \@tabclassz
    \@classiv == \@tabclassiv
    \\ == \@tabularcrrc
    \@tabarray
  }
END

\endarray == BEGIN \crrc} } $ } END

\beginarray == if next char = [ then \@array else \@array[c] fi

\beginarray[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

```

```

\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 L^AT_EX 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{\crr\egroup\egroup}
```

```
\endtabular* 160 \def\endtabular{\crr\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\z0}%
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\z0 \@@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
\arraycolsep
    5 -> \hskip \arraycolsep \@addamp \hskip
    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 \@xexpast S *0x\@@ END

\@xexpast 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 == \@xexpast
    else \reserved@b == \@xexnoop
    fi
  \expandafter \reserved@b \reserved@a S3 \@@
END
End of historical LATEX 2.09 comments.

\@xexnoop
214 \def\@xexnoop #1\@@{ }

\@expast
215 \def\@expast#1{\@xexpast #1*0x\@@}

\@xexpast
216 \def\@xexpast#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\@xexpast
223   \else
224     \let\reserved@b\@xexnoop
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{\@firstamptrue\@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
  \tabcolsep 314 \newdimen\arraycolsep
\arrayrulewidth 315 \newdimen\tabcolsep
\doublerulesep 316 \newdimen\arrayrulewidth
 317 \newdimen\doublerulesep

\arraystretch
 318 \def\arraystretch{1} % Default value.

\@arstrutbox
  \@arstrut 319 \newbox\@arstrutbox
 320 \def\@arstrut{%
 321 \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}

\@arrayrule
 322 \def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
 323 \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}

\@testpatch
 324 \def\@testpach#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}

\hline
 334 \def\hline{%
 335 \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
 336 \reserved@a\@xhline}

\@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}}

\vline
 342 \def\vline{\vrule \@width \arrayrulewidth}

\cline
\@cline The old LATEX 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 ltplain.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 cname and a register. It is declared in compatibility mode.

`\multispan` Modify `\multispan` slightly from its plain T_EX definition to allow more efficient code sharing with `\multicolumn`. Also share a count register with `\multiput`.

```
\@multispan  
  \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

68 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` `[\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):*

```

\unitlength      = value of dimension argument
\@wholewidth     = current line width
\@halfwidth      = half of current line width
\@linefont       = font for drawing lines
\@circlefont     = 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 L^AT_EX 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 \one \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 L^AT_EX 2.09 comments (not necessarily accurate any more):

```

\dashbox{D}(X,Y) ==
BEGIN
leave vertical mode
\hb@xt@ Opt {
\baselineskip := Opt
\lineskip := Opt
%% HORIZONTAL DASHES
\@dashdim := X * \unitlength
\@dashcnt := \@dashdim + 200 % to prevent roundoff error
\@dashdim := D * \unitlength
\@dashcnt := \@dashcnt / \@dashdim
if \@dashcnt is odd
then \@dashdim := Opt
\@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 L^AT_EX 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 \@ne \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 \@ne }}\@tempcnta\z@
187 \put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
188 \do{\copy\@dashbox\advance\@tempcnta \@ne }}%
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 \@ne \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]` : Same 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 \hbext@ 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 L^AT_EX 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 </2kernel>
241 <latexrelease>\IncludeInRelease{2016/03/31}%
242 <latexrelease> \@ovhlinetrue}%
243 <latexrelease> {Avoid almost zero length leaders}%
244 <*2kernel | latexrelease>

```

```

\if@ovvline Tests whether horizontal or vertical lines are needed.
\if@ovhline 245 \newif\if@ovvline \@ovvlinetrue
        246 \newif\if@ovhline \@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 \@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 </2kernel | latexrelease>
286 <latexrelease>\EndIncludeInRelease
287 <latexrelease>\IncludeInRelease{0000/00/00}%
288 <latexrelease> \if@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: “\thinlines”
363 \let\@linefnt\tenln
364 \let\@circlefnt\tencirc
365 \@wholewidth\fontdimen8\tenln
366 \@halfwidth .5\@wholewidth

```

68.1 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
  @sc := @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@ < @sc
    DO \@xdim := ((\count@*\@xa + \@xb) / @sc) * \count@
        \@ydim := ((\count@*\@ya + \@yb) / @sc) * \count@
        plot pt with relative coords (\@xdim,\@ydim)
        \count@ := \count@+1
    OD
  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\@covdy
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 </2ekernel>
421 <*2ekernel | 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\thinlines
435 \MakeRobust\vector
436 </2ekernel | 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\thinlines
452 (latexrelease)\kernel@make@fragile\vector
453 (latexrelease)
454 (latexrelease)\EndIncludeInRelease
455 <*2ekernel>
456 </2ekernel>

```

File G

lthm.dtx

69 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

70 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 (*2kernel)
2 \message{title,}
```

70.1 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 (/2kernel)
4 (*2kernel | 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,}

```

70.2 Sectioning

```

\@secpenalty
36 \newcount\@secpenalty
37 \@secpenalty = -300

```

`\ifnoskipsec` 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\ifnoskipsec \@noskipsectrue

```

`\@startsection` The `\@startsection{<name>}{<level>}{<indent>}{<beforeskip>}{<afterskip>}{<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

beforekip: 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
  \@tempkipa := BEFORESKIP
  @afterindent := T
  IF \@tempkipa < 0 THEN \@tempkipa := -\@tempkipa
                        @afterindent := F
  FI
  IF @nobreak = true
    THEN \everypar == null
    ELSE \addpenalty{\@secpenalty}
         \addvspace{\@tempkipa}
  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
        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 \@changfrom but was moved here to allow commands such as \MakeUppercase to be used at the end of #6.

```
65     #6{%
66       \@changfrom{\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 @nobreak :=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 \nobreak
91   \vskip \@tempkipa
92   \@afterheading
93   \else
94
95   \@nobreakfalse
96   \global\xnoskipsectrue
97   \everypar{%
98     \if@noskipsec
99     \global\xnoskipsecfalse
100    {\setbox\z@\lastbox}%
101    \clubpenalty\M
102    \begingroup \svsechd \endgroup
103    \unskip
104    \@tempkipa #1\relax
105    \hskip -\@tempkipa
106    \else
107    \clubpenalty \@clubpenalty
108    \everypar{}}%
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   \@tempkipa #3\relax
114   \ifdim \@tempkipa>\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}}}}

70.2.1 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,}

```

70.3 Table of Contents etc.

70.3.1 Convention

`\tf@{foo}` = file number for output for table foo. The file is opened only if `@filesw = true`.

70.3.2 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 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 </2kernel | 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 <*/2kernel)

```

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 </2kernel)

```

File I

lfloat.dtx

71 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.

71.1 Floating Environments

```
1 (*2kernel)
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.

`\c@topnumber` : Number of floats allowed at the top of a column.
`\topfraction` : Fraction of column that can be devoted to floats.
`\c@dbltopnumber, \dbltopfraction`
: Same as above, but for double-column floats.
`\c@bottomnumber, \bottomfraction`
: Same as above for bottom of page.
`\c@totalnumber` : Number of floats allowed in a single column,
including in-text floats.
`\textfraction` : Minimum fraction of column that must contain text.
`\floatpagefraction`: Minimum fraction of page that must be taken
up by float page.
`\dblfloatpagefraction`
: 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
  \@capttype ==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 `\enddblfloat`.

E.g., `\figure* == \dblfloat{figure}`,

`\endfigure* == \enddblfloat`.

```

\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     \latex@error{\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 </2kernel>
35 <latexrelease>\IncludeInRelease{2015/01/01}%
36 <latexrelease> \{\@xfloat}{Check float options}%
37 <*2kernel | latexrelease>
38 \def\@xfloat #1[#2]{%
39 \nodocument
40 \def \@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             \latexerror{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@\@capttype \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 \@capttype {#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@\@capytype \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 </2kernel>
206 <latexrelease>\IncludeInRelease{2015/01/01}%
207 <latexrelease>          {\end@dblfloat}{float order in 2-column}%
208 <*2kernel | 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 </2kernel | 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 <*2kernel>
```

```

\@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 \ifx\@floatplacement\@fpmin \else \fi

\@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)

```

When making two column float area, look for floats with 1sp depth.

```
283 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
284 \global\@dbltoproom \dbltopfraction\@colht
285 \@textmin \@colht
286 \advance \@textmin -\@dbltoproom
287 \fpmmin \dblfloatpagefraction\textheight
288 \fptop \@dblftop
289 \fpsep \@dblfpsep
290 \fpbot \@dblfpbot
\f@depth is used in \@testwrongwidth to look for either column or dbl-
column floats. A value of 1sp signals the latter. Because of this setting here,
\@dblfloatplacment needs to be called inside a group which is a questionable
design.
291 \def\f@depth{1sp}}%
292 (/2ekernel | latexrelease)
293 (latexrelease)\EndIncludeInRelease
294 (latexrelease)\IncludeInRelease{0000/00/00}%
295 (latexrelease) {\@dblfloatplacement}{float order in 2-column}%
296 (latexrelease)\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
297 (latexrelease) \global \@dbltopnum \c@dbltopnumber
298 (latexrelease) \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses \@textmin to locally store the amount of extra room in the column.

```
299 (latexrelease) \@textmin \@colht
300 (latexrelease) \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
301 (latexrelease) \fpmmin \dblfloatpagefraction\textheight
302 (latexrelease) \fptop \@dblftop
303 (latexrelease) \fpsep \@dblfpsep
304 (latexrelease) \fpbot \@dblfpbot
305 (latexrelease)}%
306 (latexrelease)\EndIncludeInRelease
307 (*2ekernel)
```

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

MARGINAL NOTES:

Marginal notes use the same mechanism as floats to communicate with the \output routine. Marginal notes are distinguished from floats by having a negative placement specification. The command \marginpar [LTEXT]{RTEXT} generates a marginal note in a parbox, using LTEXT if it's on the left and RTEXT if it's on the right. (Default is RTEXT = LTEXT.) It uses the following parameters.

\marginparwidth : Width of marginal notes.

\marginparsep : Distance between marginal note and text.
the page layout to determine how to move the marginal
note into the margin. E.g., \@leftmargin skip ==

```

\hskip -\marginparwidth \hskip -\marginparsep .
\marginparpush : Minimum vertical separation between \marginpar's

```

Marginal notes are normally put on the outside of the page if @mparswitch = true, and on the right if @mparswitch = false. The command `\reversemarginpar` reverses the side where they are put. `\normalmarginpar` undoes `\reversemarginpar`. These commands have no effect for two-column output.

SURPRISE: if two marginal notes appear on the same line of text, then the second one could appear on the next page, in a funny position.

```

\marginpar [LTEXT]{RTEXT} ==
BEGIN
  if hmode then \bsphack
    \floatpenalty := -10002
  else \floatpenalty := -10003
  fi
  if inner
    then LaTeX Error: 'Not in outer paragraph mode.'
    \floatpenalty := 0
  else if \@freelist has two elements:
    then get \@marbox, \@currbox from \@freelist
    \count\@marbox :=G -1
    else \floatpenalty := 0
    LaTeX Error: 'Too many unprocessed floats'
    \@currbox, \@marbox := \@tempboxa %%use \def
  fi
  fi
  if optional argument
    then %% \@xmpar ==
    \savemarbox\@marbox{LTEXT}
    \savemarbox\@currbox{RTEXT}
  else %% \@ympar ==
    \savemarbox\@marbox{RTEXT}
    \box\@currbox :=G \box\@marbox
  fi
  \@xympar
END

\reversemarginpar == BEGIN \@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@nobreak`; otherwise this command will be redundant.

```

346 \def \@marginparreset {%
347     \reset@font
348     \normalsize
349 %     \let@if@nobreak\iffalse
350 %     \let@if@noskipsec\iffalse
351 %     \@setnobreak
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,}

```

71.2 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 (latexrelease)\IncludeInRelease{2015/01/01}%
384 (latexrelease)           {\textsubscript}{\textsubscript}%
385 (*2kernel | latexrelease)
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 | latexrelease)
391 (latexrelease)\EndIncludeInRelease
392 (latexrelease)\IncludeInRelease{0000/00/00}%
393 (latexrelease)           {\textsubscript}{\textsubscript}%
394 (latexrelease)\let\textsubscript\@undefined
395 (latexrelease)\let@textsubscript\@undefined
396 (latexrelease)\EndIncludeInRelease
397 (*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

ltxidnglo.dtx

72 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 L^AT_EX 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 L^AT_EX 2.09 comments.

```

1 (*2kernel)
2 \message{index,}

```

`\makeindex`

```
3 \def\makeindex{%
4   \newwrite\@indexfile
5   \immediate\openout\@indexfile=\jobname.idx
6   \def\index{\@bsphack\begingroup
7     \@sanitize
8     \@windex}\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
```

`\@windex`

```
13 \def\@windex#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 </2kernel>
```

File K

ltbibl.dtx

73 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.

`\bibliography` `\bibliography{\file1,file2, ...,filen}` : specifies the bibdata files. Writes a `\bibdata` entry on the `.aux` file and tries to read in `mainfile.bbl`.

`\bibliographystyle` `\bibliographystyle{style}` : Writes a `\bibstyle` entry on the `.aux` file.

`thebibliography` The `thebibliography` environment is a list environment. To save the use of an extra counter, it should use `enumiv` as the item counter. Instead of using `\item`, items in the bibliography are produced by the following commands:

`\bibitem{<name>}` : Produces a numbered entry cited as `<name>`.

`\bibitem[<label>]{<name>}` : Produces an entry labeled by `<Label>` and cited by `<name>`.

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.

`\cite` Entries are cited by the command `\cite{<name>}`.

`\nocite` `\nocite{< citations >}` puts information on the `.aux` file that causes `BIBTEX` to include the `{< citations >}` list in the bibliography, but puts nothing in the text.

`\nocite{*}` is special: it tells `BIBTEX` to put the whole of a collection of references into the bibliography.

```
1 (*2kernel)
2 \message{bibliography,}
```

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

PARAMETERS

`\@cite` : A macro such that `\@cite{LABEL1,LABEL2}{NOTE}` produces the output for a `\cite[NOTE]{FOO1,FOO2}` command,

where entry `FOOi` is defined by `\bibitem[LABELi]{FOOi}`. The switch `@tempswa` is true if the optional `NOTE` argument is present.

The default definition is :

```
\@cite{LABELS}{NOTE} ==
  BEGIN [LABELS
    IF @tempswa = T THEN , NOTE FI
  ]
```

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@FOO` : The name or number of the reference created by `\cite{FOO}`
E.g., if `\cite{FOO} -> [17]` , then `\b@FOO -> 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 Arsenau. 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 `LATEX 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 `LATEX`, missing all citations done with `\nocite`. Thus we do only generate an error message and leave the fix for a `LATEX 2ε` successor.

```

40 \ifx\@onlypreamble\document

```

Since we are after `\begin{document}` we can do the citations:

```

41   \@for\@citeb:=#1\do{%
42     \edef\@citeb{\expandafter\@firstofone\@citeb}}%

```

```

43     \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
44     \@ifundefined{b@\@citeb}{\G@refundefinedtrue
45       \@latex@warning{Citation ‘\@citeb’ undefined}}{}%
46   \else

```

But before `\begin{document}` we raise an error message:

```

47   \@latex@error{Cannot be used in preamble}\@eha

```

Without the compatibility problems we could fix the problem as follows:

```

48   % \AtBeginDocument{\nocite{#1}}
49   \fi
50   \@esphack}

```

Since `\nocite{*}` should not produce a warning about undefined citation keys (see PR 557), we need to set the control sequence ‘`b@*`’ to something other than `\relax`. As a result `\cite{*}` will not warn either (but that never worked with `LATEX` in the first place).

```

51 \expandafter\let\csname b@*\endcsname\@empty

```

73.1 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 </2kernel>

```

File L

ltpage.dtx

74 Page styles and related commands

74.1 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.

74.2 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.)

74.3 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 (*2kernel)

`\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}%

```

```

21 (latexrelease)           {\markboth}{Make commands robust}%
22 (latexrelease)
23 (latexrelease)\kernel@make@fragile\markboth
24 (latexrelease)\kernel@make@fragile\markright
25 (latexrelease)
26 (latexrelease)\EndIncludeInRelease
27 <*2kernel)

\@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 LATεX's marks without setting a TεX mark <whatsit>.
52 \def\@themark{}{}

\mark Test versions of LATεX 2ε initialised TεX'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 TεX's parameters to their normal finicky values.
64 \DeclareRobustCommand\fussy{%
65 \emergencystretch\z@
66 \tolerance 200%
67 \hfuzz .1\p@
68 \vfuzz\hfuzz}

\overfullrule LATεX default is no overfull box rule. Changed by document class option.
69 \overfullrule 0pt
70 </2kernel)

```

File M

ltoutput.dtx

75 Output Routine

75.1 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 (defx)\begingroup
2 (defx)\makeatletter
3 (defx)\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: $\log_2 [(\backslash\text{count } I)/32]$ is the bit number corresponding to the type of float I. To see if there is any float in $\backslash\text{LIST}$ having the same type as float I, you run $\backslash\text{@bitor}$ with

$\backslash\text{NUM} = [(\backslash\text{count } I)/32] * 32.$

```
\@bitor\NUM\LIST ==
BEGIN
  @test :=G false
  { \@elt \CTR == if \NUM <> 0 then
                    if \count\CTR / \NUM is odd
                    then @test := true          fi fi
  \LIST
  }
END
```

$\backslash\text{@cons}\backslash\text{LIST}\backslash\text{NUM}$: Globally sets $\backslash\text{LIST} := \backslash\text{LIST} * \backslash\text{@elt } \backslash\text{NUM}$

```
\@cons\LIST\NUM ==
BEGIN { \@elt == \relax
        \LIST :=G \LIST \@elt \NUM
      }
```

BOX LISTS FOR FLOAT-PLACEMENT ALGORITHMS

$\backslash\text{@freelist}$: List of empty boxes for placing new floats.
 $\backslash\text{@toplist}$: List of floats to go at top of current column.
 $\backslash\text{@midlist}$: List of floats in middle of current column.
 $\backslash\text{@botlist}$: List of floats to go at bottom of current column.
 $\backslash\text{@deferlist}$: List of floats to go after current column.
 $\backslash\text{@dbltoplist}$: List of double-col. floats to go at top of current page.
 $\backslash\text{@dbldeferlist}$: List of double-column floats to go on subsequent pages.

FLOAT-PLACEMENT ALGORITHMS

$\backslash\text{@addtobot}$: Tries to put insert $\backslash\text{@currbox}$ on $\backslash\text{@botlist}$.

Called only when:

- * $\backslash\text{ht } \text{BOX} < \backslash\text{@colroom}$
- * type of $\backslash\text{@currbox}$ not on $\backslash\text{@deferlist}$
- * $\backslash\text{@colnum} > 0$
- * $\text{@insert} = \text{false}$

If it succeeds, then:

- * sets @insert true
- * decrements $\backslash\text{@botroom}$ by $\backslash\text{ht } \text{BOX}$
- * decrements $\backslash\text{@botnum}$ and $\backslash\text{@colnum}$ 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 (/2kernel)
24 (latexrelease)\IncludeInRelease{2015/10/01}%
25 (latexrelease)      {\bx@ZZ}{Extended float list}%
26 (*2kernel | latexrelease)
27 \let\@elt\newinsert
28 (*2kernel)
29 \def\@freelist{%
30   \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
31   \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
32   \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
33   \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
34 \@freelist
35 (/2kernel)
36 \ifx\numexpr\@undefined\else
37 \def\reserved@a{%
38   \@elt\bx@S\@elt\bx@T\@elt\bx@U\@elt\bx@V
39   \@elt\bx@W\@elt\bx@X\@elt\bx@Y\@elt\bx@Z
40   \@elt\bx@AA\@elt\bx@BB\@elt\bx@CC\@elt\bx@DD\@elt\bx@EE
41   \@elt\bx@FF\@elt\bx@GG\@elt\bx@HH\@elt\bx@II\@elt\bx@JJ
42   \@elt\bx@KK\@elt\bx@LL\@elt\bx@MM\@elt\bx@NN
43   \@elt\bx@OO\@elt\bx@PP\@elt\bx@QQ\@elt\bx@RR
44   \@elt\bx@SS\@elt\bx@TT\@elt\bx@UU\@elt\bx@VV
45   \@elt\bx@WW\@elt\bx@XX\@elt\bx@YY\@elt\bx@ZZ}
46 \reserved@a
47 \def\@elt{\noexpand\@elt\noexpand}
48 \edef\@freelist{\@freelist\reserved@a}
49 \fi
50 \let\reserved@a\relax
51 \let\@elt\relax
52 (/2kernel | latexrelease)
53 (latexrelease)\EndIncludeInRelease
54 (latexrelease)\IncludeInRelease{0000/00/00}%
55 (latexrelease)      {\bx@ZZ}{Extended float list}%
56 (latexrelease)\def\@freelist{%
57 (latexrelease)   \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
58 (latexrelease)   \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
59 (latexrelease)   \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
60 (latexrelease)   \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
61 (latexrelease)   \insecunt=234
62 (latexrelease)\EndIncludeInRelease
63 (*2kernel)

64 \gdef\@toplist{}
65 \gdef\@botlist{}
66 \gdef\@midlist{}
67 \gdef\@currlist{}
```

```
68 \gdef\@deferlist{}
69 \gdef\@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
\@beginvibox in the .dvi file.
```

The stuff in the box should not add any typeset material to the page when it is unboxed.

```
86 \newbox\@beginvibox
87 \DeclareRobustCommand \AtBeginDvi [1]{%
88   \global \setbox \@beginvibox
89     \vbox{\unvbox \@beginvibox #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 </2ekernel>

```

`\onecolumn`

```

141 (*2ekernel | fltrace)
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 </2ekernel | fltrace)
151 (latexrelease)\IncludeInRelease{2017/04/15}%
152 (latexrelease) \newpage}{Check depth of page}%
153 (*2ekernel | latexrelease | fltrace)
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 T_EX. 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 L^AT_EX 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 L^AT_EX 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 \@cclv`.

The last box which is removed is the box put there by the double-penalty mechanism. The `\unskip` then removes the `\topskip` which is put there since the box is the first on the page.

```
302     \global \setbox\@holdpg \vbox{%
303             \unvbox\@holdpg
304             \unvbox\@cclv
```

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\@currlist{%
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 </2ekernel | fltrace)

```

`\@testwrongwidth` Test if the float box has the wrong width when trying to place it into some area.
`\f@depth` (Actually the test is for a conventional depth setting rather than for the width of the float. For that reason the box depth was explicitly tailored when the float was created).

```

345 <latexrelease>\IncludeInRelease{2015/01/01}%
346 <latexrelease>           {\@testwrongwidth}{float order in 2-column}%
347 <*2ekernel | 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 </2ekernel | 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@fcolmade \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)    \@whilesw\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)    \@whilesw\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 </2ekernel)

```

\@makespecialcolbox This implements certain variations in page-makeup.

```

522 (*2ekernel | 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 </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 bothe 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 </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 (/2kernel | fltrace)

```

```

\@texttop These do nothing as a default.
\@textbottom 572 (*2kernel)
573 \let \@texttop \relax
574 \let \@textbottom \relax

```

```

\@resetactivechars RmS 93/09/06: added hook to protect against certain active characters in the
\@activechar@info output routine. Default checks are for active space and end-of-line.

```

```

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 `\@writsetup`.

```
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 </2kernel | 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\@ifnewlist\ifnewlist
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\@oddfont
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> \@begin@dv
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 <*2kernel>

```

`\@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 `\@cflb` 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 </2ekernel>

```

```

\@startcolumn
\@startdblcolumn

```

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 < *2ekernel | 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 
```

This one does not need to set \@colht.

```

793 
```

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 
```

```

829 (latexrelease | fltrace)}%
830 (latexrelease | fltrace)\EndIncludeInRelease
831 (*2ekernel | fltrace)

\@tryfcolumn Now tests if its list is empty before any further exertion.
832 \def \@tryfcolumn #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}

\@sdblcolet
854 \def\@sdblcolet#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 </2kernel>
873 <latexrelease>\IncludeInRelease{2015/01/01}{\@xtryfc}%
874 <latexrelease>                                {float order in 2-column}%
875 <*2kernel | 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 </2kernel | latexrelease>
893 <latexrelease>\EndIncludeInRelease
894 <latexrelease>\IncludeInRelease{0000/00/00}{\@xtryfc}%
895 <latexrelease>                                {float order in 2-column}%
896 <latexrelease>\def\@xtryfc #1{%
897   \@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 <*2kernel>
```

\@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 >\@fpmi
921     \global\@fcolmadetrue
```

```

922 \else
923 \cons\@failedlist #1%
924 \fi
925 \endgroup
926 \if@colmade
927 \let\@elt\@gobble
928 \fi}

```

\@ztryfc

```

929 </2kernel>
930 <latexrelease>\IncludeInRelease{2015/01/01}{@ztryfc}%
931 <latexrelease> \float order in 2-column}%
932 <*2kernel | 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%
    not in fixfloats?
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 </2kernel | 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<\sist@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 (*2kernel | latexrelease | fltrace | flafter)
1069 \def \@addtocurcol {%
1070 (*trace)
1071   \fl@trace{***Start addtocurcol}%
1072 
1073   \@insertfalse
1074   \@setfloattypecounts
1075   \ifnum \@fpstype=8
1076 (*trace)
1077   \fl@trace{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
1078 
1079   \else
1080   \ifnum \@fpstype=24
1081 (*trace)
1082   \fl@trace{fpstype p only (addtocurcol): \the \@fpstype = 24?}%
1083 
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 
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 
1096   \ifdim \@textmin>\@reqcolroom
1097   \reqcolroom \@textmin
1098 (*trace)
1099   \fl@trace{ORD? textmin being used}%
1100 
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 
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 \advspace 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 (*!2kernel | fltrace | latexrelease)
1172 (*trace)
1173             \fl@trace{not here: sent to addtotoporbot}%
1174 (/trace)
1175             \@addtotoporbot
1176 (/!2kernel | fltrace | latexrelease)
1177 (*!2kernel&!fltrace&!latexrelease)
1178 (*trace)
1179             \fl@trace{not here: sent to addtobot}%
1180 (/trace)
1181             \@addtobot
1182 (/!2kernel&!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         \@resetfhps
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>     \@setfloastypecounts
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     </2kernel | 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> \beginingroup
1472     (*trace)
1473     <latexrelease | fltrace> \fl@trace{***Start addtonextcol}%
1474     </trace>
1475     <latexrelease | fltrace> \@insertfalse
1476     <latexrelease | fltrace> \@setfloatypecounts
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 
```

```
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 
```

```
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 
```

```
1624      \fi
1625      \fi
1626      \if@tempswa
1627      \@bitor \@currtype \@deferlist
1628 (*trace)
1629      \fl@trace{(dbl)deferlist: \@deferlist: (before)}%
1630 
```

not in fixfloats?

```
1631      \@testwrongwidth \@currbox
1632      \if@test
1633 (*trace)
1634      \fl@trace{type already on list: (dbl)defer}%
1635 
```

```
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 
```

```

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<\sist@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) \@setfloatypecounts
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) \@tempwafalse
1702 (latexrelease | fltrace) \ifdim \@dbltoproom>\ht\@currbox

```

```

1703 <latexrelease | fltrace>          \@tempswatruue
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<\sift@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>          \@tempswatruue
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 \@currtype \@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<\sist@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 | ftrace)}%
1810 (latexrelease | ftrace)\EndIncludeInRelease

```

`\@addmarginpar`

```

1811 (*2kernel)
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}}

```

75.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for L^AT_EX₂ε.

```
\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>

```

75.1.2 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_EX₂ε.

```
\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 (*fltrace)
\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.¹¹

```

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

```

¹¹This 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   {\PackageWarningNoLine
1960     {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
1961       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` 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 \</2kernel>

```

`\@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@\@capttype \endcsname}%
  \edef \reserved@a {\the\@temptokena}%
  \@onelevel@sanitize \reserved@a
  \edef \@fps {\@fps\reserved@a}%
}

```

```

1990 \<2kernel | fltrace>
1991 \def \@fpsadddefault {%
1992 \<trace>
1993 \fl@trace{fps changed from: \@fps}%
1994 \</trace>
1995 \edef \@fps {\@fps\csname fps@\@capttype \endcsname}%
1996 \@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 \@currtype \count\@currbox
2002 \@fpstype \count\@currbox
2003 \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
2004 \advance \@fpstype -\@currtype
2005 \<trace>
2006 \fl@trace{(mod 32) fpstype: \the \@fpstype}%
2007 \fl@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 \</2ekernel | fltrace>

```

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```

2029 (*2ekernel)
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<\sift@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 `@tempwa` 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<\sift@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 </trace>
2138     \fi
2139     \fi
2140 (*trace)
2141     \else
2142     \fl@trace{Fail---no room at 2nd test of colroom
2143             (flcheckspace \string#1 \string#2)}%
2144 </trace>
2145     \fi
2146 }
2147 </2ekernel | fltrace>

```

`\@flupdates` This updates everything when a float is placed.

```

2148 (*2ekernel)
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 </2ekernel>

```

Interesting facts about float mechanisms past and present, together with a summary of various features, some unresolved:

1. The value `\textfraction` does not affect the processing of doublecol floats: this seems sensible, but should be documented.
2. `\twocolumn` floatplacement was wrong: dbl not needed, ord needed.
3. `\floatplacement` was not called after `\startdblcol` or `\topnewpage`. This has been changed; it is clearly a bug fix.
4. The use `\@topnewpage` when `\dblfigrule` is non-trivial produced a rule in the wrong place. This has been fixed by not using `\dblfigrule` when processing the ‘float’ from `\@topnewpage`.
5. If the specifier was just h and the float could not be put here, it went on the deferlist and stayed there until a clearpage. It now gets changed to a ‘th’: this is only an error-recovery action, putting just h or !h should be deprecated.
6. `\@dblmaxsep` was ‘the maximum of `\dblfloatsep` and `\dbltextfloatsep`’. But it was never used! Now gone completely, like `\@maxsep`.

7. After an h float is put on a page, it was counted as text when applying the `\textfraction` test; this is possibly too big a change although it is a bug fix?
8. Two consecutive h floats are separated by twice `\intertextsep`: this could be changed to one by use of `\addvspace`, OK? Note that it would also mean that less space is put in if an h float immediately follows other spaces. This is also possibly too big a change, at least for compatibility mode? Or it may be simply wrong! It has not been changed.
9. Now `\@addtocurcol` checks first for just p fps. I think that this is an increase in efficiency, but maybe the coding should be made even more efficient.
10. `\@tryfcolumn` now tests if the list is empty first, otherwise lots of wasted time! Thus this test has been removed from `\@startcolumn`. As Frank pointed out, this makes `\@startcolumn` less efficient. But it is now the same as `\@startdblcolumn`: I can see no reason why they should be different, but which is best?
11. Why is `\@colroom` set in `\@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
```

```

2165 \@fpmin -\maxdimen
2166 \let \@testfp \@gobble
2167 \@tryfcolumn #1%
2168 \endgroup
2169 (*trace)
2170 \if@fcolmade
2171 \fl@trace{PAGE: in \string\clearpage
2172 \if@twocolumn ---twocolumn\fi---}%
2173 \fl@trace{----- float column/page completed from \string#1}%
2174 \fi
2175 </trace>
2176 }

2177 (latexrelease | fltrace)\EndIncludeInRelease
2178 (latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
2179 (latexrelease | fltrace) {\@makefcolumn}{negative height floats}%
2180 (latexrelease | fltrace)\def\@makefcolumn #1{%
2181 (latexrelease | fltrace) \begingroup
2182 (latexrelease | fltrace) \@fpmin \z@
2183 (latexrelease | fltrace) \let \@testfp \@gobble
2184 (latexrelease | fltrace) \@tryfcolumn #1%
2185 (latexrelease | fltrace) \endgroup
2186 (*trace)
2187 (latexrelease | fltrace) \if@fcolmade
2188 (latexrelease | fltrace) \fl@trace{PAGE: in \string\clearpage
2189 (latexrelease | fltrace) \if@twocolumn ---twocolumn\fi---}%
2190 (latexrelease | fltrace) \fl@trace{----- float column/page completed
2191 (latexrelease | fltrace) from \string#1}%
2192 (latexrelease | fltrace) \fi
2193 </trace>
2194 (latexrelease | fltrace)}
2195 (latexrelease | fltrace)\EndIncludeInRelease
2196 </2ekernel | fltrace | latexrelease>

```

This will line up the last baselines in the two columns provided they are constructed in the normal way: i.e. ending in a skip of minus the original depth, with `\@textbottom` adding nothing.

Thus again it is essential for `\@textbottom` to have depth 0pt.

```

2197 (latexrelease | fltrace)\IncludeInRelease{2015/01/01}%
2198 (latexrelease | fltrace) {\@outputdblcol}{2 column marks}%
2199 (*2ekernel | fltrace | latexrelease)

```

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```

2200 \def\@outputdblcol{%
2201 \if@firstcolumn
2202 \global\@firstcolumnfalse

Save the left column

2203 \global\setbox\@leftcolumn\copy\@outputbox
2204 (fltrace) \fl@trace{PAGE: first column boxed}%

```

Remember the marks from the first column

```

2205 \splitmaxdepth\maxdimen
2206 \vbadness\maxdimen

```

In case of `\enlargethispage` we will have infinite negative glue at the bottom of the page (coming from `\vss`) and that will earn us an error message if we `\vsplit` to get at the marks. So we need to remove the last glue (if any) at the end of `\@outputbox` as we are only interested in marks that change doesn't matter.

```
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 \LaTeX marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```
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 (fltrace) \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 (fltrace) \fl@trace{PAGE: two column page completed}%
2234 \begingroup
2235 \dblfloatplacement
2236 \startdblcolumn
2237 \@whilesw\if@fcolmade \fi{\@outputpage
2238 (fltrace) \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)     \@whilesw\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 (/2kernel | fltrace | latexrelease)

```

75.1.3 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

- `\c@topnumber` This counter holds the maximum number of floats that can appear at the top of a text page or column.
2288 (*2ekernel)
2289 `\newcount\c@topnumber`
2290 `\setcounter{topnumber}{2}`
- `\topfraction` 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 `\newcommand\topfraction{.7}`
- `\c@bottomnumber` This counter holds the maximum number of floats that can appear at the bottom of a text page or column.
2292 `\newcount\c@bottomnumber`
2293 `\setcounter{bottomnumber}{1}`
- `\bottomfraction` This macro holds the maximum proportion (as a decimal number) of a text page or column that can be occupied by floats at the bottom.
2294 `\newcommand\bottomfraction{.3}`
- `\c@totalnumber` This counter holds the maximum number of floats that can appear on any text page or column.
2295 `\newcount\c@totalnumber`
2296 `\setcounter{totalnumber}{3}`
- `\textfraction` This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text.
2297 `\newcommand\textfraction{.2}`
- `\floatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page or column that must be occupied by floating objects before a ‘float page’ is produced.
2298 `\newcommand\floatpagefraction{.5}`
- `\c@dbltopnumber` This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page.
2299 `\newcount\c@dbltopnumber`
2300 `\setcounter{dbltopnumber}{2}`
- `\dbltopfraction` This macro holds the maximum proportion (as a decimal number) of a two-column text page that can be occupied by double-column floats at the top.
2301 `\newcommand\dbltopfraction{.7}`
- `\dblfloatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page that must be occupied by double-column floating objects before a ‘double-column float page’ is produced.
2302 `\newcommand\dblfloatpagefraction{.5}`

Floats on a text page

`\floatsep` `\textfloatsep` `\intertextsep` 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.

`\intertextsep` is the space between in-text floats and the text.

```
2303 \newskip\floatsep
2304 \newskip\textfloatsep
2305 \newskip\intertextsep
2306 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
2307 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
2308 \setlength\intertextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
```

`\dblfloatsep` `\dbltextfloatsep` When double-column floats (floating objects that span the whole `\textwidth`) are placed at the top of a text page in two-column mode, the separation between the float and the text is controlled by `\dblfloatsep` and `\dbltextfloatsep`. They are rubber lengths.

`\dblfloatsep` is the space between adjacent double-column floats placed at the top of the text page.

`\dbltextfloatsep` is the space between the main text and double-column floats at the top of the page.

```
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` `\@fpsep` `\@fpbot` When floating objects are placed on a separate page or column, called a ‘float page’, the layout of the page is controlled by these parameters, which are rubber lengths.

At the top of the page `\@fptop` is inserted; typically this supplies some stretchable whitespace. At the bottom of the page `\@fpbot` is inserted. Between adjacent floats `\@fpsep` is inserted.

These parameters are used for all floating objects on a ‘float page’ in one-column mode, and for single-column floats in two-column mode.

Note that at least one of the two parameters `\@fptop` and `\@fpbot` should contain a `plus ...fil` so as to fill the remaining empty space.

```
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` `\@dblfpsep` `\@dblfpbot` Double-column ‘float pages’ in two-column mode use similar parameters.

```
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 </2ekernel>
```

File N

ltxclass.dtx

76 Introduction

This file implements the following declarations, which replace `\documentstyle` in $\text{\LaTeX} 2_{\epsilon}$ documents.

Note that old documents containing `\documentstyle` will be run using a compatibility option—thus keeping everyone happy, we hope!

The overall idea is that there are two types of ‘style files’: ‘class files’ which define elements and provide a default formatting for them; and ‘packages’ which provide extra functionality. One difference between $\text{\LaTeX} 2_{\epsilon}$ and $\text{\LaTeX} 2.09$ is that $\text{\LaTeX} 2_{\epsilon}$ packages may have options. Note that options to classes packages may be implemented such that they input files, but these file names are not necessarily directly related to the option name.

77 User interface

```
\documentclass[<main-option-list>]{<class>}[<version>]
```

There must be exactly one such declaration, and it must come first. The *<main-option-list>* is a list of options which can modify the formatting of elements which are defined in the *<class>* file as well as in all following `\usepackage` declarations (see below). The *<version>* is a version number, beginning with a date in the format YYYY/MM/DD. If an older version of the class is found, a warning is issued.

```
\documentstyle[<main-option-list>]{<class>}[<version>]
```

The `\documentstyle` declaration is kept in order to maintain upward compatibility with $\text{\LaTeX} 2.09$ documents. It is similar to `\documentclass`, but it causes all options in *<main-option-list>* that the *<class>* does not use to be passed to `\RequirePackage` after the options have been processed. This maintains compatibility with the 2.09 behaviour. Also a flag is set to indicate that the document is to be processed in $\text{\LaTeX} 2.09$ compatibility mode. As far as most packages are concerned, this only affects the warnings and errors \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.

77.1 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.

78 Class and Package interface

78.1 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.

78.2 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.

78.3 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 <code>\@ifpackageloaded{<package>}{<true>}{<false>}</code>.</p> <p>To find out if a package has already been loaded with a version equal to or more recent than <code><version></code>, use <code>\@ifpackagelater{<package>}{<version>}{<true>}{<false>}</code>.</p> <p>To find out if a package has already been loaded with at least the options <code><options></code>, 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).

78.4 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 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.

78.5 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.

79 Implementation

```

1 (*2kernel)
\if@compatibility The flag for compatibility mode.
2 \newif\if@compatibility
\@documentclasshook The hook called after the first \documentclass command. By default this checks
to see if \@normalsize is undefined, and if so, sets it to \normalsize.
3 \def\@documentclasshook{%
4   \ifx\@normalsize\undefined
5     \let\@normalsize\normalsize
6   \fi
7 }
\@declaredoptions This list is automatically built by \DeclareOption. 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 \ds<option> commands are executed. All local <option>s which are not declared will be processed in the order defined by the optional argument of \documentclass or \usepackage.
8 \let\@declaredoptions\@empty
\@classoptionslist 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{\@ifloaded\@pkgextension}
41 \def\@ifclassloaded{\@ifloaded\@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.\@current\endcsname{#1}%
157   \ifx\@current\@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.\@current\endcsname{#1}%
172 <latexrelease>   \ifx\@current\@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 `ltxfinal.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 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\@xprocessoptions\@processoptions}
230 \@onlypreamble\ProcessOptions

231 \def\@processoptions{%
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   \@processoptions}
243 \@onlypreamble\@processoptions

244 \def\@xprocessoptions{%
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 \@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 list 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_EX 2.09 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 \if2kernel)
277 \iflatexrelease)\IncludeInRelease{2017/01/01}%
278 \iflatexrelease)          {\ExecuteOptions}{Spaces in \ExecuteOptions}%
279 \if*2kernel|latexrelease)
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 </2kernel | 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 <*2kernel>
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 L^AT_EX 2_ε 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     \@needsf@rmat
357   {}}
358 \@onlypreamble\@needsformat

359 \def\@needsf@rmat[#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\@needsf@rmat
```

`\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 </2kernel>
380 <latexrelease>\IncludeInRelease{2017/01/01}%
381 <latexrelease>   {\@fileswith@ptions}{ifx tests in \@fileswith@ptions}%
382 <*2kernel | 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@options

```

```

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@ptions\@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@ptions\@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\@clsextension\let\LoadClass\@twoloadclasserror\fi
485   \@popfilename
486   \@reset@ptions}%
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

```

79.1 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.

```

\@begindocumenthook
\@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
\AtBeginDocument 509 \def\AtEndOfPackage{%
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

```

79.2 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{\@tempwattrue\@fileswtrue
572   \ifnextchar[\filec@ntents@opt\filec@ntents
573 }
574 \namedef{filecontents*}{\@tempwafalse\@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   \edef\@fortmp{\zap@space#1 \@empty}%
579   \@for\reserved@a:=\@fortmp\do{%
580     \ifcsname filec@ntents@\reserved@a\endcsname
581       \csname filec@ntents@\reserved@a\endcsname
582     \else
583       \@latex@error{Unknown filecontents option \reserved@a}%
584       {Valid options are force (or overwrite), nosearch, noheader}%
585     \fi}%
586   \filec@ntents
587 }

```

Option force) (or overwrite) changes the overwriting switch

```

588 \let\filec@ntents@force\@fileswfalse
589 \let\filec@ntents@overwrite\@fileswfalse % alternative name

```

and option noheader the preamble switch (which is equivalent to using the star form of the environment).

```

590 \let\filec@ntents@noheader\@tempwafalse

```

Option nosearch only checks the current directory not the how T_EX tree for the existence of the file to write.

```

591 \def\filec@ntents@nosearch{%
592   \let\filec@ntents@checkdir\@currdir
593   \def\filec@ntents@where{in current directory}}

```

By default we search the whole tree:

```

594 \let\filec@ntents@checkdir\@empty
595 \def\filec@ntents@where{exists on the system}

```

```

596 \begingroup%
597 \@tempcnta=1
598 \loop

```

```

599 \catcode\@tempcnta=12 %
600 \advance\@tempcnta\@ne %
601 \ifnum\@tempcnta<32 %
602 \repeat %
603 \catcode'\*=11 %
604 \catcode'\^^M\active%
605 \catcode'\^^L\active\let^^L\relax%
606 \catcode'\^^I\active%

607 \gdef\filec@ntents#1{%
608 \set@curr@file{\filec@ntents@checkdir#1}%
609 \edef\q@curr@file{\expandafter\quote@name\expandafter{\@curr@file}}%

    LuaTeX has more writes (and 18 is safe here).

610 \chardef\reserved@c\ifx\directlua\undefined 15 \else 127 \fi%
611 \openin\@inputcheck\q@curr@file \space %
612 \ifeof\@inputcheck%
613 \@latex@warning@no@line%
614 {Writing file '\currdir\@curr@file'}%

615 \ch@ck7\reserved@c\write\relax%
616 \immediate\openout\reserved@c\q@curr@file\relax%
617 \else%

618 \if@filesw%
619 \@latex@warning@no@line%
620 {File '\@curr@file' already \filec@ntents@where.\MessageBreak%
621 Not generating it from this source}%
622 \let\write\@gobbletwo%
623 \let\closeout\@gobble%
624 \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!

```

625 \edef\reserved@a{#1}%
626 \edef\reserved@a{\detokenize\expandafter{\reserved@a}}%
627 \edef\reserved@b{\detokenize\expandafter{\jobname}}%
628 \ifx\reserved@a\reserved@b%
629 \@fileswtrue%
630 \else%
631 \edef\reserved@b{\reserved@b\detokenize{.tex}}%
632 \ifx\reserved@a\reserved@b
633 \@fileswtrue%
634 \fi%
635 \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.

```

636 \ch@ck7\reserved@c\write\relax%
637 \if@filesw% % Foul ... trying to overwrite \jobname!
638 \@latex@error{Trying to overwrite '\jobname.tex'}{You can't %
639 write to the file you are reading from!\MessageBreak%
640 Data is written to screen instead.}%
641 \else%

```

```

642     \@latex@warning@no@line%
643     {Writing or overwriting file ‘\@currdir\@curr@file’}%
644     \immediate\openout\reserved@c\q@curr@file\relax%
645     \fi%
646     \fi%
647     \fi%

```

Closing the \@inputcheck is done here to avoid having to do this in each branch.

```

648     \closein\@inputcheck%
649     \if@tempwa%

650     \immediate\write\reserved@c{%
651     \@percentchar\@percentchar\space%
652     \expandafter\@gobble\string\LaTeX2e file ‘\@curr@file’^^J%
653     \@percentchar\@percentchar\space generated by the %
654     ‘\@currenvir’ \expandafter\@gobblefour\string\newenvironment^^J%
655     \@percentchar\@percentchar\space from source ‘\jobname’ on %
656     \number\year/\two@digits\month/\two@digits\day.^^J%
657     \@percentchar\@percentchar}%
658     \fi%
659     \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.

```

660     \count@ 128\relax%
661     \loop%
662     \catcode\count@ 11\relax%
663     \advance\count@ \@ne%
664     \ifnum\count@<\@ccclvi%
665     \repeat%

666     \edef\E{\@backslashchar end\string{\@currenvir\string}}%
667     \edef\reserved@b{%
668     \def\noexpand\reserved@b%
669     #####1\E####2\E####3\relax}%
670     \reserved@b{%
671     \ifx\relax###3\relax%

```

There was no \end{filecontents}

```

672     \immediate\write\reserved@c{##1}%
673     \else%

```

There was a \end{filecontents}, so stop this time.

```

674     \edef^^M{\noexpand\end{\@currenvir}}%
675     \ifx\relax##1\relax%
676     \else%

```

Text before the \end, write it with a warning.

```

677     \@latex@warning{Writing text ‘##1’ before %
678     \string\end{\@currenvir}\MessageBreak as last line of \@curr@file}%
679     \immediate\write\reserved@c{##1}%
680     \fi%
681     \ifx\relax##2\relax%
682     \else%

```

Text after the `\end`, ignore it with a warning.

```
683     \latexwarning{%
684       Ignoring text ‘#2’ after \string\end{\@currenvir}}%
685     \fi%
686     \fi%
687     ^^M}%

688 \catcode‘^^L\active%
689 \let\L\@undefined%
690 \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
691 \catcode‘^^I\active%
692 \let\I\@undefined%
693 \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
694 \catcode‘^^M\active%
695 \edef^^M##1^^M{%
696   \noexpand\reserved@b##1\E\E\relax}}%
697 \endgroup%

698 </2kernel | latexrelease>
699 <latexrelease>\EndIncludeInRelease
700 <latexrelease>\IncludeInRelease{0000/00/00}%
701 <latexrelease>           {\filec@ntents}{Spaces in file names + optional arg}%
702 <latexrelease>
703 <latexrelease>\let\filec@ntents@opt      \@undefined
704 <latexrelease>\let\filec@ntents@force   \@undefined
705 <latexrelease>\let\filec@ntents@overwrite \@undefined
706 <latexrelease>\let\filec@ntents@noheader \@undefined
707 <latexrelease>\let\filec@ntents@nosearch \@undefined
708 <latexrelease>\let\filec@ntents@checkdir \@undefined
709 <latexrelease>\let\filec@ntents@where   \@undefined
710 <latexrelease>
711 <latexrelease>\begingroup%
712 <latexrelease>\@tempcnta=1
713 <latexrelease>\loop
714 <latexrelease> \catcode\@tempcnta=12 %
715 <latexrelease> \advance\@tempcnta\@ne %
716 <latexrelease>\ifnum\@tempcnta<32 %
717 <latexrelease>\repeat %
718 <latexrelease>\catcode‘*=11 %
719 <latexrelease>\catcode‘^^M\active%
720 <latexrelease>\catcode‘^^L\active\let^^L\relax%
721 <latexrelease>\catcode‘^^I\active%
722 <latexrelease>
723 <latexrelease>\gdef\filec@ntents#1{%
724 <latexrelease> \openin\@inputcheck#1 %
725 <latexrelease> \ifeof\@inputcheck%
726 <latexrelease>   \latexwarning@no@line%
727 <latexrelease>     {Writing file ‘\@currdir#1’}%
728 <latexrelease>   \chardef\reserved@c15 %
729 <latexrelease>   \ch@ck7\reserved@c\write%
730 <latexrelease>   \immediate\openout\reserved@c#1\relax%
731 <latexrelease> \else%
732 <latexrelease>   \closein\@inputcheck%
733 <latexrelease>   \latexwarning@no@line%
734 <latexrelease>     {File ‘#1’ already exists on the system.\MessageBreak%
```

```

735 (latexrelease)           Not generating it from this source}%
736 (latexrelease)   \let\write@gobbletwo%
737 (latexrelease)   \let\closeout@gobble%
738 (latexrelease)   \fi%
739 (latexrelease)   \if@tempswa%
740 (latexrelease)   \immediate\write\reserved@c{%
741 (latexrelease)     \@percentchar\@percentchar\space%
742 (latexrelease)     \expandafter\@gobble\string\LaTeX2e file '#1'^^J%
743 (latexrelease)     \@percentchar\@percentchar\space generated by the %
744 (latexrelease)     '@currentvir' \expandafter\@gobblefour\string\newenvironment^^J%
745 (latexrelease)     \@percentchar\@percentchar\space from source '\jobname' on %
746 (latexrelease)     \number\year/\two@digits\month/\two@digits\day.^^J%
747 (latexrelease)     \@percentchar\@percentchar}%
748 (latexrelease)   \fi%
749 (latexrelease)   \let\do\makeother\dospecials%
750 (latexrelease)   \count@ 128\relax%
751 (latexrelease)   \loop%
752 (latexrelease)     \catcode\count@ 11\relax%
753 (latexrelease)     \advance\count@ \@ne%
754 (latexrelease)     \ifnum\count@<\ccclvi%
755 (latexrelease)     \repeat%
756 (latexrelease)     \edef\E{\@backslashchar end\string{\@currentvir\string}}%
757 (latexrelease)     \edef\reserved@b{%
758 (latexrelease)       \def\noexpand\reserved@b%
759 (latexrelease)         ###1\E###2\E###3\relax}%
760 (latexrelease)     \reserved@b{%
761 (latexrelease)       \ifx\relax##3\relax%
762 (latexrelease)         \immediate\write\reserved@c{##1}%
763 (latexrelease)       \else%
764 (latexrelease)         \edef^^M{\noexpand\end{\@currentvir}}%
765 (latexrelease)         \ifx\relax##1\relax%
766 (latexrelease)         \else%
767 (latexrelease)           \@latex@warning{Writing text '#1' before %
768 (latexrelease)             \string\end{\@currentvir}MessageBreak as last line of #1}%
769 (latexrelease)           \immediate\write\reserved@c{##1}%
770 (latexrelease)         \fi%
771 (latexrelease)         \ifx\relax##2\relax%
772 (latexrelease)         \else%
773 (latexrelease)           \@latex@warning{%
774 (latexrelease)             Ignoring text '##2' after \string\end{\@currentvir}}%
775 (latexrelease)         \fi%
776 (latexrelease)       \fi%
777 (latexrelease)     ^^M}%
778 (latexrelease)
779 (latexrelease)   \catcode'\^^L\active%
780 (latexrelease)   \let\L\@undefined%
781 (latexrelease)   \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
782 (latexrelease)   \catcode'\^^I\active%
783 (latexrelease)   \let\I\@undefined%
784 (latexrelease)   \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
785 (latexrelease)   \catcode'\^^M\active%
786 (latexrelease)   \edef^^M##1^^M{%
787 (latexrelease)     \noexpand\reserved@b##1\E\E\relax}}%
788 (latexrelease)\endgroup%

```

```

789 (latexrelease)\EndIncludeInRelease
790 (*2ekernel)

791 \begingroup
792 \catcode'\|=\catcode'\%
793 \catcode'\%=12
794 \catcode'\*=11
795 \gdef\@percentchar{%}
796 \gdef\endfilecontents{|
797   \immediate\closeout\reserved@o
798   \def\T##1##2##3{|
799   \ifx##1\@undefined\else
800     \@latex@warning@no@line{##2 has been converted to Blank ##3e}|
801   \fi}|
802 \T\L{Form Feed}{Lin}|
803 \T\I{Tab}{Spac}|
804 \immediate\write\@unused{}}
805 \global\let\endfilecontents*\endfilecontents

```

We no longer prevent the code to be used after begin document (no rollback needed for this change).

```

806 %\@onlypreamble\filecontents
807 %\@onlypreamble\endfilecontents
808 %\@onlypreamble\filecontents*
809 %\@onlypreamble\endfilecontents*
810 \endgroup
811 %\@onlypreamble\filec@ntents

```

80 Package/class rollback mechanism

```

812 </2ekernel)
813 (*2ekernel | 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.

```

814 (*tracerollback)
815 %\let\pkgcls@debug\typeout
816 \let\pkgcls@debug\@gobble
817 </tracerollback)

```

`\requestedLaTeXdate` The macro (!) `\requestedLaTeXdate` holds the globally requested rollback date (via `latexrelease`) or zero if no such request was made.

```

818 \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.

```
819 \ifx\pkgcls@targetdate\@undefined
820 \newcount\pkgcls@targetdate
821 \fi
822 \let\pkgcls@targetlabel\@empty
823 \def\pkgcls@innerdate{\maxdimen}
```

`\pkgcls@candidate` When looping through the `\DeclareRelease` declarations we record if the release
`\pkgcls@releasedate` 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.

```
824 \let\pkgcls@candidate\@empty
825 \let\pkgcls@releasedate\@empty
```

`\load@onefilewithoptions` the best place to add the rollback code is at the point where `\@onefilewithoptions`
`\@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.

```
826 \ifx\load@onefilewithoptions\@undefined
827 \let\load@onefilewithoptions\@onefilewithoptions
828 \def\@onefilewithoptions#1[#2][#3]#4{%
```

First a bit of tracing normally disabled.

```
829 (*traceroollback)
830 \pkgcls@debug{--- File loaded request (\noexpand\usepackage or ...)}%
831 \pkgcls@debug{\@spaces 1: #1}%
832 \pkgcls@debug{\@spaces 2: #2}%
833 \pkgcls@debug{\@spaces 3: #3}%
834 \pkgcls@debug{\@spaces 4: #4}%
835 </traceroollback)
```

Two of the arguments are needed later on in error/warning messages so we save them.

```
836 \def\pkgcls@name{#1}% % for info message
837 \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`.

```
838 \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:

```
839 \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.

```

840 \begingroup
841 \edef\reserved@a{%
842   \endgroup
843   \unexpanded{\load@onefilewithoptions#1[#2]}%
844   [\pkgcls@mindate]%
845   \unexpanded{#4}}%
846 \reserved@a
847 }
848 \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>	<code>1</code>	<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.

```
849 \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.

```

850   \ifx\@nil#1\@nil
851     \pkgcls@targetdate\requestedLaTeXdate\relax
852     \let\pkgcls@targetlabel\@empty
853     \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.

```
854   \else
855     \pkgcls@parse@date@arg@#1=\@nil\relax
856   \fi
857 }
```

The actual parsing work then happens in `\pkgcls@parse@date@arg@`:

```
858 \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.

```
859   \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.

```
860     \ifx\@nil#1\@nil
861       \@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 =.

```
862     \ifnum \pkgcls@targetdate=\z@
863       \pkgcls@targetdate\@ne
864       \def\pkgcls@innerdate{\maxdimen}%
865       \pkgcls@parse@date@arg@version#2%
866     \else
867       \edef\pkgcls@innerdate{\the\pkgcls@targetdate}%
868     \fi
869     \let\pkgcls@mindate\@empty
870   \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`):

```
871     \requestedLaTeXdate\relax
872     \let\pkgcls@targetlabel\@empty
873     \def\pkgcls@innerdate{\maxdimen}%
874     \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.

```
875     \ifnum \pkgcls@targetdate > \z@
876       \ifnum \@parse@version0#1//00\@nil > \pkgcls@targetdate
877         \@latex@warning@no@line{Suspicious rollback/min-date date given\MessageBreak
878           A minimal date of #1 has been specified for
879           \@cls@pkg\MessageBreak '\pkgcls@name'.\MessageBreak
880           But this is in conflict
881           with a rollback request to \requestedpatchdate}
```

```

882     \fi
883     \fi
884     \fi
885 }

```

Strip off the trailing = and assign the version name to `\pkgcls@targetlabel`.

```

886 \def\pkgcls@parse@date@arg@version#1={%
887   \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.

```

888 \def\DeclareRelease#1#2#3{%
889   \ifnum\pkgcls@targetdate>\z0 % some sort of rollback request
890 (*traceroollback)
891   \pkgcls@debug{---\string\DeclareRelease:}%
892   \pkgcls@debug{\@spaces 1: #1}%
893   \pkgcls@debug{\@spaces 2: #2}%
894   \pkgcls@debug{\@spaces 3: #3}%
895 </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.

```

896   \ifx\@nil#2\@nil
897     \ifnum\pkgcls@targetdate=\@ne % named request
898     \def\reserved@a{#1}%
899     \ifx\pkgcls@targetlabel\reserved@a
900     \pkgcls@use@this@release{#3}{}%
901 (*traceroollback)
902     \else
903     \pkgcls@debug{Label doesn't match}%
904 </traceroollback)
905     \fi
906 (*traceroollback)
907     \else
908     \pkgcls@debug{Date request: ignored}%
909 </traceroollback)
910     \fi
911     \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.

```

912     \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.

```

913     \ifnum\@parse@version#2//00\@nil
914     >\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.

```

915         \ifx\pkgcls@candidate\@empty
916             \pkgcls@rollbackdate@error{#2}%
917             \pkgcls@use@this@release{#3}{#2}%
918         \else
919             \pkgcls@use@this@release\pkgcls@candidate
920                                 \pkgcls@releasedate
921         \fi
922     \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.

```

923         \def\pkgcls@candidate{#3}%
924         \def\pkgcls@releasedate{#2}%
925 (*tracereollback)
926         \pkgcls@debug{New candidate: #3}%
927 (/tracereollback)
928     \fi
929 \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.

```

930     \def\reserved@a{#1}%
931     \ifx\pkgcls@targetlabel\reserved@a
932         \pkgcls@use@this@release{#3}{#2}%
933 (*tracereollback)
934     \else
935         \pkgcls@debug{Label doesn't match}%
936 (/tracereollback)
937     \fi
938 \fi
939 \fi
940 \fi
941 }

```

`\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.

```
942 \def\pkgcls@use@this@release#1#2{%
```

Before that we record the selection made inside the transcript.

```
943     \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 bypassed¹² and then we finally load the correct release.

¹²The 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.

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!`

```
944 \pkgcls@targetdate\z@
945 \@@input #1\relax
946 \endinput
947 }
```

`\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.

```
948 \def\pkgcls@show@selection#1#2{%
949 (*traceroollback)
950 \pkgcls@debug{Result: use #1}%
951 (/traceroollback)
952 \GenericInfo
953 {\@spaces\@spaces\space}{Rollback for
954 \@cls@pkg\space'\@currname' requested ->
955 \ifnum\pkgcls@targetdate>\@ne
956 date
957 \ifnum\requestedLaTeXdate=\pkgcls@targetdate
958 \requestedpatchdate
959 \else
960 \expandafter\@gobble\pkgcls@arg
961 \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.

```
962 Best approximation is
963 \else
964 version '\pkgcls@targetlabel'.\MessageBreak
965 This corresponds to
966 \fi
967 \ifx\@nil#2\@nil
968 a special release%
969 \else
970 the release introduced on #2%
971 \fi
972 \@gobble}%
973 }
```

`\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.

```
974 \def\pkgcls@rollbackdate@error#1{%
975 \@latex@error{Suspicious rollback date given}%
976 {The \@cls@pkg\space'\@currname' claims that it
977 came into existence on #1 which\MessageBreak
```

```

978     is after your requested rollback date --- so
979     something is wrong here.\MessageBreak
980     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.

```

981 \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).
982 \ifnum\pkgcls@targetdate>\z@ % some sort of rollback request
983 (*traceroollback)
984 \pkgcls@debug{---DeclareCurrentRelease}%
985 \pkgcls@debug{ 1: #1}%
986 \pkgcls@debug{ 2: #2}%
987 /traceroollback)

```

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`.

```

988 \ifnum\pkgcls@targetdate>\one % a date request
989 \ifnum\@parse@version#2//00\@nil
990 >\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).

```

991 \ifx\pkgcls@candidate\@empty
992 \pkgcls@rollbackdate@error{#2}%
993 \else
994 \pkgcls@use@this@release\pkgcls@candidate
995 \pkgcls@releasedate
996 \fi

```

Otherwise the current file is the right release, so we record that in the transcript and then carry on.

```

997 \else
998 \pkgcls@show@selection{current version}{#2}%
999 \fi
1000 \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.

```

1001 \def\reserved@a{#1}%
1002 \ifx\pkgcls@targetlabel\reserved@a
1003 \pkgcls@show@selection{current version}{#2}%
1004 \else
1005 \@latex@error{Requested version '\pkgcls@targetlabel' for
1006 \@cls@pkg\space'\@currname' is unknown}\@ehc
1007 \fi
1008 \fi
1009 \fi
1010 }

```

`\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.

```

1011 \DeclareRobustCommand\IfTargetDateBefore[1]{%
1012   \ifnum\pkgcls@innerdate <%
1013     \expandafter\@parse@version\expandafter0#1//00\@nil
1014     \typeout{Exclude code introduced on #1}%
1015     \expandafter\@firstoftwo
1016   \else
1017     \typeout{Include code introduced on #1}%
1018     \expandafter\@secondoftwo
1019   \fi
1020 }

1021 </2ekernel | latexreleasefirst>

```

81 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```

1022 <*/afterpreamble>
1023 \NeedsTeXFormat{LaTeX2e}
1024 \ProvidesPackage{pkgindoc}
1025   [1994/10/20 v1.1 Package Interface in Document (DPC)]
1026 \def\reserved@a#1\do\@classoptionslist#2\do\filecontents#3\relax{%
1027   \gdef\@preamblecmds{#1#3}}
1028 \expandafter\reserved@a\@preamblecmds\relax
1029 </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 </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 </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

ltxuatex.dtx

82 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^ATeX 2_ε kernel level plus as a loadable file which can be used with plain TeX and L^ATeX.

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^ATeX 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^ATeX 2_ε kernel did not provide any functionality for the extended allocation area).

83 Core TeX functionality

The commands defined here are defined for possible inclusion in a future L^ATeX format, however also extracted to the file `ltxuatex.tex` which may be used with older L^ATeX formats, and with plain TeX.

```
\newattribute \newattribute{<attribute>}
  Defines a named \attribute, indexed from 1 (i.e. \attribute0 is never defined).
  Attributes initially have the marker value -"7FFFFFFF ('unset') set by the engine.
\newcatcodetable \newcatcodetable{<catcodetable>}
  Defines a named \catcodetable, indexed from 1 (\catcodetable0 is never assigned).
  A new catcode table will be populated with exactly those values assigned by IniTeX
  (as described in the LuaTeX manual).
\newluafunction \newluafunction{<function>}
  Defines a named \luafunction, indexed from 1. (Lua indexes tables from 1 so
  \luafunction0 is not available).
  \newwhatsit \newwhatsit{<whatsit>}
  Defines a custom \whatsit, indexed from 1.
\newluabytecode \newluabytecode{<bytecode>}
```

	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.

84 Plain T_EX interface

The `luatex` interface may be used with plain T_EX using `\input{ltluatex}`. This inputs `ltluatex.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.

85 Lua functionality

85.1 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>-"7FFFFFFF</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 \TeX count register to manage their allocations. The standard names are those defined above for access from \TeX , e.g. “e@alloc@attribute@count, but these can be adjusted by defining the variable $\langle type \rangle_count_name$ before loading `ltxlua.tex`, for example

```
local attribute_count_name = "attributetracker"
require("ltxlua")
```

would use a \TeX `\count` (`\countdef`'d token) called `attributetracker` in place of “e@alloc@attribute@count.

85.2 Lua access to \TeX register numbers

`registernumber` `luatexbase.registernumber($\langle name \rangle$)`
Sometimes (notably in the case of Lua attributes) it is necessary to access a register *by number* that has been allocated by \TeX . This package provides a function to look up the relevant number using Lua \TeX '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 Lua \TeX 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`).

85.3 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.

85.4 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($\langle name \rangle$,metatype,$\langle default \rangle$)</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.

86 Implementation

```

1 (*2kernel | tex | latexrelease)
2 (2kernel | latexrelease)\ifx\directlua\@undefined\else

```

86.1 Minimum LuaTeX version

LuaTeX has changed a lot over time. In the kernel support for ancient versions is not provided: trying to build a format with a very old binary therefore gives some information in the log and loading stops. The cut-off selected here relates to the tree-searching behaviour of `require()`: from version 0.60, LuaTeX will correctly find Lua files in the `texmf` tree without ‘help’.

```

3 (latexrelease)\IncludeInRelease{2015/10/01}
4 (latexrelease)          {\newluafunction}{LuaTeX}%
5 \ifnum\luatexversion<60 %
6   \wlog{*****}
7   \wlog{* LuaTeX version too old for ltuatex support *}
8   \wlog{*****}
9   \expandafter\endinput
10 \fi

```

86.2 Older L^ATeX/Plain T_EX setup

```

11 (*tex)

```

Older L^ATeX formats don’t have the primitives with ‘native’ names: sort that out. If they already exist this will still be safe.

```

12 \directlua{tex.enableprimitives("",tex.extraprimitives("luatex"))}
13 \ifx\@alloc\@undefined

```

In pre-2014 L^ATeX, or plain T_EX, load `etex.sty,src`.

```

14 \ifx\documentclass\@undefined
15   \ifx\loccount\@undefined
16     \input{etex.src}%
17   \fi
18   \catcode'\@=11 %

```

```

19   \outer\expandafter\def\csname newfam\endcsname
20   {\alloc@8\fam\chardef\et@xmaxfam}
21   \else
22     \RequirePackage{etex}
23     \expandafter\def\csname newfam\endcsname
24     {\alloc@8\fam\chardef\et@xmaxfam}
25     \expandafter\let\expandafter\new@mathgroup\csname newfam\endcsname
26     \fi

```

86.2.1 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_E_X-specific code, but does not define the correct range for Lua_T_E_X.

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@cclvi\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 ltuatex.)

```

36 % \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}

```

End of proposed changes to `etex.src`

86.2.2 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 `ltxltx.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>

```

86.3 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}`

86.4 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

```

86.5 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\mne\e@alloc@top
179 }

```

```
180 \e@alloc@luafunction@count=\z@
```

86.6 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@
```

86.7 Lua bytecode registers

`\newluabytecode` 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\newluabytecode#1{%
193   \e@alloc\luabytecode\e@alloc@chardef
194   \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
195 }
196 \e@alloc@bytecode@count=\z@
```

86.8 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@
```

86.9 Lua loader

Load the Lua code at the start of every job. For the conversion of T_EX 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 (2kernel)\everyjob\expandafter{%
208 (2kernel) \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("lualatex")}
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\@f@encoding
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\@f@encoding\encodingdefault
```

```

263 \expandafter\let\csname ver@luaotfload.sty\endcsname\fmtversion
264 </2kernel, latexrelease>
265 <latexrelease>\fi
266 <2kernel> }
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 <2kernel | latexrelease>\fi
273 </2kernel | tex | latexrelease>

```

86.10 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

```

86.11 Lua module utilities

86.11.1 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

```

86.11.2 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

```

86.12 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

```

86.13 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

```

86.14 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

```

86.15 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

```

86.16 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

```

86.17 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
```

86.18 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.

86.18.1 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, reverselist = 1, 2, 3, 4, 5
475 local types = {
476     list      = list,
477     data      = data,
478     exclusive = exclusive,
479     simple    = simple,
480     reverselist = reverselist,
481 }
```

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 Lua_T_EX. (Some undocumented callbacks are omitted as they are to be removed.)

```
482 local callbacktypes = callbacktypes or {
```

Section 8.2: file discovery callbacks.

```
483 find_read_file      = exclusive,
484 find_write_file     = exclusive,
485 find_font_file      = data,
486 find_output_file    = data,
487 find_format_file    = data,
488 find_vf_file        = data,
489 find_map_file       = data,
490 find_enc_file       = data,
491 find_pk_file        = data,
492 find_data_file      = data,
493 find_opentype_file  = data,
494 find_truetype_file  = data,
495 find_type1_file     = data,
496 find_image_file     = data,
```

```
497 open_read_file      = exclusive,
498 read_font_file      = exclusive,
499 read_vf_file        = exclusive,
500 read_map_file       = exclusive,
501 read_enc_file       = exclusive,
502 read_pk_file        = exclusive,
503 read_data_file      = exclusive,
504 read_truetype_file  = exclusive,
505 read_type1_file     = exclusive,
506 read_opentype_file  = exclusive,
```

Not currently used by luatex but included for completeness. may be used by a font handler.

```
507 find_cidmap_file   = data,
508 read_cidmap_file    = exclusive,
```

Section 8.3: data processing callbacks.

```
509 process_input_buffer = data,
510 process_output_buffer = data,
511 process_jobname      = data,
```

Section 8.4: node list processing callbacks.

```
512 contribute_filter   = simple,
513 buildpage_filter    = simple,
514 build_page_insert   = exclusive,
515 pre_linebreak_filter = list,
516 linebreak_filter    = exclusive,
517 append_to_vlist_filter = exclusive,
518 post_linebreak_filter = reverselist,
519 hpack_filter        = list,
520 vpack_filter        = list,
521 hpack_quality       = list,
522 vpack_quality       = list,
523 pre_output_filter   = list,
524 process_rule        = exclusive,
```

```

525 hyphenate           = simple,
526 ligaturing         = simple,
527 kerning            = simple,
528 insert_local_par   = simple,
529 pre_mlist_to_hlist_filter = list,
530 mlist_to_hlist     = exclusive,
531 post_mlist_to_hlist_filter = reverselist,
532 new_graf           = simple,

```

Section 8.5: information reporting callbacks.

```

533 pre_dump           = simple,
534 start_run          = simple,
535 stop_run           = simple,
536 start_page_number = simple,
537 stop_page_number  = simple,
538 show_error_hook   = simple,
539 show_warning_message = simple,
540 show_error_message = simple,
541 show_lua_error_hook = simple,
542 start_file         = simple,
543 stop_file          = simple,
544 call_edit          = simple,
545 finish_synctex    = simple,
546 wrapup_run        = simple,

```

Section 8.6: PDF-related callbacks.

```

547 finish_pdf_file   = data,
548 finish_pdf_page   = data,
549 page_objnum_provider = data,
550 page_order_index  = data,
551 process_pdf_image_content = data,

```

Section 8.7: font-related callbacks.

```

552 define_font       = exclusive,
553 glyph_info        = exclusive,
554 glyph_not_found   = exclusive,
555 glyph_stream_provider = exclusive,
556 make_extensible   = exclusive,
557 font_descriptor_objnum_provider = exclusive,

```

```
558 }
```

```
559 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.

```

560 local callback_register = callback_register or callback.register
561 function callback.register()
562   luatexbase_error("Attempt to use callback.register() directly\n")
563 end

```

86.18.2 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.

reverselist is a specialized variant of *list* which executes functions in inverse order.

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.

```
564 local function data_handler(name)
565   return function(data, ...)
566     for _,i in ipairs(callbacklist[name]) do
567       data = i.func(data,...)
568     end
569     return data
570   end
571 end
```

Default for user-defined **data** callbacks without explicit default.

```
572 local function data_handler_default(value)
573   return value
574 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.

```
575 local function exclusive_handler(name)
576   return function(...)
577     return callbacklist[name][1].func(...)
578   end
579 end
```

Handler for list callbacks.

```
580 local function list_handler(name)
581   return function(head, ...)
582     local ret
583     local alltrue = true
584     for _,i in ipairs(callbacklist[name]) do
585       ret = i.func(head, ...)
586       if ret == false then
587         luatexbase_warning(
588           "Function '" .. i.description .. "' returned false\n"
589           .. "in callback '" .. name .. "'")
590       )
591       break
592     end
593     if ret ~= true then
594       alltrue = false
595       head = ret
596     end
597   end
598   return alltrue and true or head
599 end
600 end
```

Default for user-defined list and reverselist callbacks without explicit default.

```
601 local function list_handler_default()
602   return true
603 end
```

Handler for reverselist callbacks.

```
604 local function reverselist_handler(name)
605   return function(head, ...)
606     local ret
607     local alltrue = true
608     local callbacks = callbacklist[name]
609     for i = #callbacks, 1, -1 do
610       local cb = callbacks[i]
611       ret = cb.func(head, ...)
612       if ret == false then
613         luatexbase_warning(
614           "Function '" .. cb.description .. "' returned false\n"
615           .. "in callback '" .. name .. "'")
616       )
617       break
618     end
619     if ret ~= true then
620       alltrue = false
621       head = ret
622     end
623   end
624   return alltrue and true or head
625 end
626 end
```

Handler for simple callbacks.

```
627 local function simple_handler(name)
```

```

628 return function(...)
629   for _,i in ipairs(callbacklist[name]) do
630     i.func(...)
631   end
632 end
633 end

```

Default for user-defined `simple` callbacks without explicit default.

```

634 local function simple_handler_default()
635 end

```

Keep a handlers table for indexed access and a table with the corresponding default functions.

```

636 local handlers = {
637   [data]      = data_handler,
638   [exclusive] = exclusive_handler,
639   [list]      = list_handler,
640   [reverselist] = reverselist_handler,
641   [simple]     = simple_handler,
642 }
643 local defaults = {
644   [data]      = data_handler_default,
645   [exclusive] = nil,
646   [list]      = list_handler_default,
647   [reverselist] = list_handler_default,
648   [simple]     = simple_handler_default,
649 }

```

86.18.3 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.

```

650 local user_callbacks_defaults = {
651   pre_mlist_to_hlist_filter = list_handler_default,
652   mlist_to_hlist           = node.mlist_to_hlist,
653   post_mlist_to_hlist_filter = list_handler_default,
654 }

```

`create_callback` The allocator itself.

```

655 local function create_callback(name, ctype, default)
656   local ctype_id = types[ctype]
657   if not name or name == ""
658   or not ctype_id
659   then
660     luatexbase_error("Unable to create callback:\n" ..
661                       "valid callback name and type required")
662   end
663   if callbacktypes[name] then
664     luatexbase_error("Unable to create callback '" .. name ..
665                       "':\ncallback is already defined")
666   end
667   default = default or defaults[ctype_id]
668   if not default then

```

```

669     luatexbase_error("Unable to create callback '" .. name ..
670                       "':\ndefault is required for '" .. ctype ..
671                       "' callbacks")
672   elseif type (default) ~= "function" then
673     luatexbase_error("Unable to create callback '" .. name ..
674                       "':\ndefault is not a function")
675   end
676   user_callbacks_defaults[name] = default
677   callbacktypes[name] = ctype_id
678 end
679 luatexbase.create_callback = create_callback

```

`call_callback` Call a user defined callback. First check arguments.

```

680 local function call_callback(name,...)
681   if not name or name == "" then
682     luatexbase_error("Unable to create callback:\n" ..
683                       "valid callback name required")
684   end
685   if user_callbacks_defaults[name] == nil then
686     luatexbase_error("Unable to call callback '" .. name
687                       .. "':\nunknown or empty")
688   end
689   local l = callbacklist[name]
690   local f
691   if not l then
692     f = user_callbacks_defaults[name]
693   else
694     f = handlers[callbacktypes[name]](name)
695   end
696   return f(...)
697 end
698 luatexbase.call_callback=call_callback

```

`add_to_callback` Add a function to a callback. First check arguments.

```

699 local function add_to_callback(name, func, description)
700   if not name or name == "" then
701     luatexbase_error("Unable to register callback:\n" ..
702                       "valid callback name required")
703   end
704   if not callbacktypes[name] or
705     type(func) ~= "function" or
706     not description or
707     description == "" then
708     luatexbase_error(
709       "Unable to register callback.\n\n"
710       .. "Correct usage:\n"
711       .. "add_to_callback(<callback>, <function>, <description>)"
712     )
713   end

```

Then test if this callback is already in use. If not, initialise its list and register the proper handler.

```

714   local l = callbacklist[name]
715   if l == nil then

```

```

716     l = { }
717     callbacklist[name] = l

```

If it is not a user defined callback use the primitive callback register.

```

718     if user_callbacks_defaults[name] == nil then
719         callback_register(name, handlers[callbacktypes[name]](name))
720     end
721 end

```

Actually register the function and give an error if more than one exclusive one is registered.

```

722 local f = {
723     func      = func,
724     description = description,
725 }
726 local priority = #l + 1
727 if callbacktypes[name] == exclusive then
728     if #l == 1 then
729         luatexbase_error(
730             "Cannot add second callback to exclusive function\n" ..
731             name .. "'")
732     end
733 end
734 table.insert(l, priority, f)

```

Keep user informed.

```

735 luatexbase_log(
736     "Inserting '" .. description .. "' at position "
737     .. priority .. " in '" .. name .. "'")
738 )
739 end
740 luatexbase.add_to_callback = add_to_callback

```

`remove_from_callback` Remove a function from a callback. First check arguments.

```

741 local function remove_from_callback(name, description)
742     if not name or name == "" then
743         luatexbase_error("Unable to remove function from callback:\n" ..
744             "valid callback name required")
745     end
746     if not callbacktypes[name] or
747         not description or
748         description == "" then
749         luatexbase_error(
750             "Unable to remove function from callback.\n\n"
751             .. "Correct usage:\n"
752             .. "remove_from_callback(<callback>, <description>)"
753         )
754     end
755     local l = callbacklist[name]
756     if not l then
757         luatexbase_error(
758             "No callback list for '" .. name .. "'\n")
759     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.

```

760 local index = false
761 for i,j in ipairs(l) do
762   if j.description == description then
763     index = i
764     break
765   end
766 end
767 if not index then
768   luatexbase_error(
769     "No callback '" .. description .. "' registered for '" ..
770     name .. "'\n")
771 end
772 local cb = l[index]
773 table.remove(l, index)
774 luatexbase_log(
775   "Removing '" .. description .. "' from '" .. name .. "'."
776 )
777 if #l == 0 then
778   callbacklist[name] = nil
779   callback_register(name, nil)
780 end
781 return cb.func,cb.description
782 end
783 luatexbase.remove_from_callback = remove_from_callback

```

`in_callback` Look for a function description in a callback.

```

784 local function in_callback(name, description)
785   if not name
786     or name == ""
787     or not callbacklist[name]
788     or not callbacktypes[name]
789     or not description then
790     return false
791   end
792   for _, i in pairs(callbacklist[name]) do
793     if i.description == description then
794       return true
795     end
796   end
797   return false
798 end
799 luatexbase.in_callback = in_callback

```

`disable_callback` As we subvert the engine interface we need to provide a way to access this functionality.

```

800 local function disable_callback(name)
801   if(callbacklist[name] == nil) then
802     callback_register(name, false)
803   else
804     luatexbase_error("Callback list for '" .. name .. "' not empty")
805   end
806 end
807 luatexbase.disable_callback = disable_callback

```

`callback_descriptions` List the descriptions of functions registered for the given callback.

```
808 local function callback_descriptions (name)
809   local d = {}
810   if not name
811     or name == ""
812     or not callbacklist[name]
813     or not callbacktypes[name]
814   then
815     return d
816   else
817     for k, i in pairs(callbacklist[name]) do
818       d[k]= i.description
819     end
820   end
821   return d
822 end
823 luatexbase.callback_descriptions =callback_descriptions
```

`uninstall` Unlike at the \TeX 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!

```
824 local function uninstall()
825   module_info(
826     "luatexbase",
827     "Uninstalling kernel luatexbase code"
828   )
829   callback.register = callback_register
830   luatexbase = nil
831 end
832 luatexbase.uninstall = uninstall
```

`mlist_to_hlist` To emulate these callbacks, the “real” `mlist_to_hlist` is replaced by a wrapper calling the wrappers before and after.

```
833 callback_register("mlist_to_hlist", function(head, display_type, need_penalties)
834   local current = call_callback("pre_mlist_to_hlist_filter", head, display_type, need_penalties)
835   if current == false then
836     flush_list(head)
837     return nil
838   elseif current == true then
839     current = head
840   end
841   current = call_callback("mlist_to_hlist", current, display_type, need_penalties)
842   local post = call_callback("post_mlist_to_hlist_filter", current, display_type, need_penalties)
843   if post == true then
844     return current
845   elseif post == false then
846     flush_list(current)
847     return nil
848   end
849   return post
850 end)
851  $\langle$ /lua $\rangle$ 
```

Reset the catcode of @.

```
852 (tex)\catcode'\@=\etacatcode\relax
```

File Q

ltfinal.dtx

87 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.

87.1 Debugging

By default, L^AT_EX shows statistics:

```
1 \*2kernel)
2 \tracingstats1
```

87.2 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 \<latexrelease)\IncludeInRelease{2015/01/01}%
9 \<latexrelease) {\newmarks}{Extended Allocation}%
10 \ifx\marks\@undefined\else
11 \def\newmarks{%
12 \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\e@alloc@top}
13 \fi
14 \</2kernel | latexrelease)
15 \<latexrelease)\EndIncludeInRelease
16 \<latexrelease)\IncludeInRelease{0000/00/00}%
17 \<latexrelease) {\newmarks}{Extended Allocation}%
18 \<latexrelease)\let\newmarks\@undefined
19 \<latexrelease)\EndIncludeInRelease
20 \<*2kernel)

\newXeTeXintercharclass Allocate \XeTeXintercharclass types if xetex is active. previously defined in
\xe@alloc@intercharclass xetex.ini.
\@e@alloc@intercharclass@top 21 \</2kernel)
22 \<*2kernel | latexrelease)
23 \<latexrelease)\IncludeInRelease{2015/01/01}%
24 \<latexrelease) {\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 \e@alloc\XeTeXcharclass
34 \chardef\xe@alloc@intercharclass\m@ne\xe@alloc@intercharclass@top}
35 \fi

36 </2ekernel | latexrelease>
37 <latexrelease>\EndIncludeInRelease
38 <latexrelease>\IncludeInRelease{0000/00/00}%
39 <latexrelease> \{\newXeTeXintercharclass\}{Extended Allocation}%
40 <latexrelease> \ifx\XeTeXcharclass\@undefined
41 <latexrelease> \else
42 <latexrelease> \def\xe@alloc@#1#2#3#4#5{\global\advance#1\@ne
43 <latexrelease> \xe@ch@ck#1#4#2%
44 <latexrelease> \allocationnumber#1%
45 <latexrelease> \global#3#5\allocationnumber
46 <latexrelease> \wlog{\string#5=\string#2\the\allocationnumber}}
47 <latexrelease> \def\xe@ch@ck#1#2#3{%
48 <latexrelease> \ifnum#1<#2\else
49 <latexrelease> \errmessage{No room for a new #3}%
50 <latexrelease> \fi}
51 <latexrelease> \def\newXeTeXintercharclass{%
52 <latexrelease> \xe@alloc@\xe@alloc@intercharclass
53 <latexrelease> \XeTeXcharclass\chardef\@cclv}
54 <latexrelease> \fi
55 <latexrelease>\EndIncludeInRelease
56 *2ekernel | latexrelease>
57 <latexrelease>\IncludeInRelease{2016/02/01}%
58 <latexrelease> {\xe@alloc@intercharclass\}{Start of XeTeX class allocator}%
59 \ifx\XeTeXcharclass\@undefined
60 \else
61 \countdef\xe@alloc@intercharclass=257
62 \xe@alloc@intercharclass=\z@
63 \fi
64 </2ekernel | latexrelease>
65 <latexrelease>\EndIncludeInRelease
66 <latexrelease>\IncludeInRelease{2015/01/01}%
67 <latexrelease> {\xe@alloc@intercharclass\}{Start of XeTeX class allocator}%
68 <latexrelease> \ifx\XeTeXcharclass\@undefined
69 <latexrelease> \else
70 <latexrelease> \xe@alloc@intercharclass=\thr@@
71 <latexrelease> \fi
72 <latexrelease>\EndIncludeInRelease
73 <latexrelease>\IncludeInRelease{0000/00/00}%
74 <latexrelease> {\xe@alloc@intercharclass\}{Start of XeTeX class allocator}%
75 <latexrelease> \ifx\XeTeXcharclass\@undefined
76 <latexrelease> \else
77 <latexrelease> \newcount\xe@alloc@intercharclass
78 <latexrelease> \xe@alloc@intercharclass=\thr@@
79 <latexrelease> \fi

```

```

80 <latexrelease>\EndIncludeInRelease
81 <*2kernel>

```

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

```

87.3 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 </2kernel>
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 (*2ekernel)

```

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
```

87.4 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>
```

87.5 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
```

87.6 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`, `enctex` or `mltex`.

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 (/2ekernel)
243 (*2ekernel | latexrelease)
```

Check that a classic 8-bit tex engine is being used (LaTeX or PDFLaTeX).

```
244 (latexrelease)\IncludeInRelease{2018/04/01}%
245 (latexrelease)           {\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\@inpenc@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).

87.7 Lccodes and uccodes

We now again set up the default (T1) uc/lccodes. The lower case characters need their $\text{\textbackslash uccode}$ and $\text{\textbackslash lccode}$ 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}}

```

87.8 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 % \typeout{Applying patch file ltpatch.ltx^^J%
487 % \typeout{=====}}
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}%

```


87.11 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
`\@addtofilelist` built up so far may take up a lot of memory and so it is moved to `\reserved@a` where it will be overwritten as soon as almost any L^AT_EX 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}}%
```

87.12 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}
```

87.13 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 ...
```

87.14 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. 356
- 1989-04-10 ltfssbas.dtx v1.0a
 General: Starting with version numbers! `\ifmmode` added in `\math@group` 154
- 1989-04-10 ltfssbas.dtx v1.0b
 General: `\preload@sizes` added. 154
`\wrong@fontshape` changed to define substitution font/shape macro. 154
- 1989-04-10 ltfssini.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`. . 247
- 1989-04-14 ltfssbas.dtx v1.0c
 General: More documentation added. 154
- 1989-04-15 ltfssini.dtx v1.0b
 General: `\mathfontset` renamed to `\mathversion`. 247
- 1989-04-19 ltfssbas.dtx v1.0d
 General: Even more doc. 154
- 1989-04-21 ltfssbas.dtx v1.0e
 General: Documentation is fun! Parameters of `\define@mathalphabet` changed. 154
- 1989-04-21 ltfssini.dtx v1.0c
 General: Changed to conform to fam.tex. 247
- 1989-04-23 ltfssbas.dtx v1.0f
 General: % in `\getanddefinefonts` added. 154
- 1989-04-26 ltfssini.dtx v1.0d
 General: `\xpt` added. 247
- 1989-04-27 ltfssbas.dtx v1.0g
 General: Documentation revised. 154
- 1989-04-27 ltfssini.dtx v1.0e
 General: Definitions of L^AT_EX symbols corrected. 247
- 1989-04-29 ltfssbas.dtx v1.0h
 General: Documented problem with `\halign`, and `\noalign` 154
`\mathversion`: Test if version defined added. 164
- 1989-04-29 ltfssbas.dtx v1.0i
 General: Removed the `\halign` `\noalign` correction (wasn't bugfree) 154
- 1989-04-29 ltfssini.dtx v1.0f
 General: Corrections to L^AT_EX tabular env. added. 247
- 1989-05-01 ltfssbas.dtx v1.0j
 General: Default for `\baselinestretch` added. . . 154
- 1989-05-22 ltfssbas.dtx v1.0k
 General: Lines longer than 72 characters folded. 154
- 1989-05-22 ltfssini.dtx v1.0g
 General: Lines shortened to 72 characters 247
- 1989-09-14 ltfssbas.dtx v1.0m
 General: Global replacement: `\group` to `\mathgroup` 154
`\mathversion`: Corrected typo: `\endscname` to `\endcsname`. . 164
- 1989-11-07 ltfssini.dtx v1.0i
 General: All family, series, and shape names abbreviated. . . 247
- 1989-11-08 ltfssbas.dtx v1.0o
 General: First parameter of `\define@mathalphabet` and `\define@mathgroup` changed from string to control sequence. 154
- 1989-11-14 ltfssbas.dtx v1.0p
`\math@version`: Math version prefix 'mv@' added. 164
- 1989-11-19 ltfssbas.dtx v1.0q
`\define@newfont`: Group added. 165
`\wrong@fontshape`: Instead of calling `\family\default@family`, etc. we directly set `\f@family`, etc. 170
- 1989-11-22 ltfssbas.dtx v1.0r
`\math@version`: `\def` → `\edef` for `\math@version`. 164
- 1989-11-25 ltfssbas.dtx v1.0s
 General: All `\edef\font@name` changed to `\xdef\font@name`.

Necessary after introduction of <code>\begingroup/\endgroup</code> in v1.0q.	154	1990-01-21 ltfssstrc.dtx v1.2b <code>\use@mathgroup</code> : Macro added to allow cleaner interface.	205
<code>extra//</code> \rightarrow <code>+</code> in <code>\extra@def</code> . . .	154	1990-01-23 ltfssbas.dtx v1.2c General: <code>\no@version@warning</code> renamed to <code>\no@alphabet@error</code>	154
1989-11-26 ltfssbas.dtx v1.0t <code>\select@group:\bgroup/\egroup</code> changed to <code>\begingroup/\endgroup</code> to avoid empty Ord atom on math list.	172	Macro <code>\no@alphabet@help</code> added	154
1989-12-02 ltfssini.dtx v1.1b General: <code>\rmmath</code> renamed to <code>\mathrm</code>	247	<code>\no@alphabet@error</code> : Changed to error call	154
1989-12-03 ltfssini.dtx v1.1c General: Some internal macros renamed to make them inaccessible.	247	1990-01-25 ltfssini.dtx v1.1e <code>\nfss@text</code> : Macro added.	261
1989-12-05 ltfssbas.dtx v1.0u <code>\addto@hook:\addto@hook</code> added.	177	1990-01-27 ltfssbas.dtx v1.2d <code>\DeclarePreloadSizes</code> : Font identifier set to <code>\relax</code>	160
1989-12-05 ltfssstrc.dtx v1.0u fam.dtx <code>\every@math@size</code> : Hook <code>\every@size</code> added.	202	1990-01-28 ltfssbas.dtx v1.2e <code>\mathgroup:\newfam</code> let to <code>\new@mathgroup</code>	154
1989-12-13 ltfssstrc.dtx v1.0f <code>\use@mathgroup:\expandafter</code> added before final <code>\fi</code>	205	1990-01-28 ltfssbas.dtx v1.2f <code>\define@newfont</code> : Added call to <code>\curr@fontshape</code> macro to allow substitution.	166
1989-12-16 ltfssbas.dtx v1.1a <code>\select@group:\relax</code> in front added.	172	<code>\wrong@fontshape</code> : Warning message slightly changed.	170
Now four arguments.	172	1990-01-28 ltfssini.dtx v1.2b <code>\em</code> : Call to <code>\nomath</code> added.	259
Redefinition of alphabet now simpler.	173	1990-02-08 ltfssini.dtx v1.1g General: Protected the commands <code>\family</code> , <code>\series</code> , <code>\shape</code> , <code>\size</code> , <code>\selectfont</code> , and <code>\mathversion</code>	247
Usage of <code>'=</code> ' macro added.	173	1990-02-16 ltfssbas.dtx v1.2g General: Support for changes of <code>\baselineskip</code> without changing the size.	154
1989-12-16 ltfssstrc.dtx v1.1a <code>\selectfont</code> : Changed order of calls.	199	<code>\math@version:\nomath</code> added.	164
<code>\use@mathgroup</code> : Redefinition of alphabet now simpler.	205	1990-02-16 ltfssstrc.dtx v1.0i <code>\selectfont</code> : Changed <code>\f@size</code> to <code>\lcl@currsz</code> (see fam file).	199
Usage of <code>'=</code> ' macro added.	205	1990-02-18 ltfssstrc.dtx v1.0j General: Redefine unprotected version <code>\p@selectfont</code> instead of <code>\selectfont</code>	199
1990-01-18 ltfssstrc.dtx v1.0h General: <code>\tracingfonts</code> meaning changed.	195	1990-03-14 ltfssstrc.dtx v1.0k General: Added code for TeX3.	195
1990-01-20 ltfssbas.dtx v1.2a <code>\math@bgroup</code> : Def. placed in this file.	174	<code>\extract@font</code> : Added code for TeX3.	198
<code>\math@egroup</code> : Def. placed in this file.	174	<code>\selectfont</code> : Added code for TeX3.	199
<code>\select@group</code> : Def for alph id changed.	173	1990-03-30 ltfssbas.dtx v1.2h <code>\math@egroup</code> : Changed to have one arg.	174
1990-01-21 ltfssbas.dtx v1.2b <code>\select@group</code> : Code moved to <code>\use@mathgroup</code>	173		

1990-03-30 ltfssstrc.dtx v1.2h		1990-08-27 ltfssstrc.dtx 1.0r	
<code>\use@mathgroup</code> : Third argument removed (see <code>\math@egroup</code>).	205	<code>\type@restoreinfo</code> : Some extra tracing info.	201
1990-04-01 ltfssbas.dtx v1.2i		1990-08-27 ltfssstrc.dtx v1.0r	
General: Code added from <code>tracefnt.dtx</code>	154	<code>\getanddefine@fonts</code> : Correcting missing name after <code>\tracingon</code>	206
Support for TeX3.	154	1991-03-28 ltfssini.dtx v1.1m	
1990-04-01 ltfssstrc.dtx v1.0l		<code>\copyright</code> : Extra braces added.	261
General: Part of code moved to <code>fam.dtx</code>	195	1991-03-30 ltfssini.dtx v1.2g	
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> already defined.	196	<code>\newfont</code> : Definition added.	260
1990-04-01 ltfssstrc.dtx v1.0o		<code>\symbol</code> : Definition added.	260
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> defined removed again.	196	1991-07-24 ltmiscen.dtx LaTeX2.09	
1990-04-02 ltfssini.dtx v1.1i		<code>\@verbatim</code> : Added <code>\penalty\interlinepenalty</code> to definition of <code>\par</code> so that <code>\samepage</code> works	343
General: <code>\input</code> of files now handled by <code>docstrip</code>	247	1991-08-14 ltmath.dtx LaTeX2.09	
1990-04-05 ltfssstrc.dtx v1.0m		<code>\cases</code> : (RmS) inserted extra braces around entry for NFSS	352
<code>\selectfont</code> : Call <code>\tracingon</code> only if <code>\tracingfonts</code> greater than 3.	199	1991-08-14 ltpictur.dtx LaTeX2.09	
1990-05-05 ltfssstrc.dtx v1.0n		General: (RmS) inserted extra braces around entry for NFSS	414
<code>\selectfont</code> : <code>\tracingon</code> with new syntax.	199	1991-08-14 ltthm.dtx LaTeX2.09	
1990-06-23 ltfssini.dtx v1.1k		<code>\@endtheorem</code> : Moved <code>\itshape</code> after <code>\item</code> to make it work with NFSS	437
<code>\nfss@text</code> : Changed to <code>\mbox</code>	261	1991-08-26 ltfssini.dtx v1.1n	
1990-06-24 ltfssbas.dtx v1.2j		<code>\reset@font</code> : Macro introduced	261
<code>\DeclarePreloadSizes</code> : Missing percent added.	159	1991-08-26 ltmiscen.dtx LaTeX2.09	
1990-06-24 ltfssstrc.dtx v1.0o		<code>\@verbatim</code> : <code>\@@par</code> added	343
<code>\baselinestretch</code> : Moved to <code>tracefnt.dtx</code>	202	1991-08-26 ltpictur.dtx LaTeX2.09	
<code>\getanddefine@fonts</code> : Adding tracing code.	206	<code>\endpicture</code> : (RmS & FmI) extra boxing level around <code>\@picbox</code> to guard against unboxing in math mode (proposed by John Hobby)	413
<code>\Macro</code> moved from <code>fam.dtx</code>	206	1991-08-26 ltplain.dtx LaTeX2.09	
Adding debug code.	206	<code>\tracingall</code> : Added <code>\errorcontextlines=\maxdimen</code> , suggested by J. Schrod	30
<code>\use@mathgroup</code> : Tracing code added.	205	1991-09-29 ltboxes.dtx LaTeX2.09	
1990-06-30 ltfssbas.dtx v1.2l		<code>\@mpfootnotetext</code> : (RmS) added <code>\reset@font</code>	386
<code>\showhyphens</code> : Macro added.	175	1991-09-29 ltfloat.dtx LaTeX2.09	
1990-06-30 ltfssstrc.dtx v1.0p		<code>\@footnotetext</code> : (RmS) added <code>\reset@font</code>	467
<code>\use@mathgroup</code> : Added <code>\relax</code> after math group number.	205	1991-09-29 ltmath.dtx LaTeX2.09	
1990-07-07 ltfssstrc.dtx v1.0q		<code>\@eqnnum</code> : RmS: <code>\reset@font</code> added.	355
<code>\getanddefine@fonts</code> : Group number added to tracing.	206	1991-09-29 ltsect.dtx LaTeX2.09	
<code>\math@egroup</code> : Tracing code added.	205	<code>\@dottedtocline</code> : (RmS) added <code>\reset@font</code> for page number	448
<code>\use@mathgroup</code> : Group number added to tracing.	205		

- 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. 472
- 1991-11-01 ltfloat.dtx LaTeX2.09
`\footnote:` (RmS) Added
`\let\protect\noexpand` in
`\footnote`, `\footnotemark`,
and `\footnotetext`, since
`\xdef` is used 467
- 1991-11-04 ltlists.dtx LaTeX2.09
`\makelabel:` (RmS) added default
definition for `\makelabel`, to
produce an error message. 373
- 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 472
- 1991-11-13 ltbibl.dtx LaTeX2.09
`\@bibitem:` Changed counter
`enumi` to `enumiv`, as it says in
the comment above 472
- 1991-11-21 ltffsini.dtx v1.1o
`\reset@font:` Added extra braces
for robustness. 261
Changed to protected version of
macro. 261
- 1991-11-22 ltfloat.dtx LaTeX2.09
`\footnote:` (RmS) Added
`\let\protect\noexpand` in
`\@xfootnote`,
`\@xfootnotemark`, and
`\@xfootnotetext` 467
- 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. 373
- 1991-11-27 ltffsbas.dtx v1.3a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily` etc. 154
- 1991-11-27 ltffsini.dtx v1.2a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily` etc. 247
- 1992-01-06 ltffsini.dtx v1.2c
General: added `slitex` code 247
- 1992-01-10 ltbibl.dtx LaTeX2.09
`\@bibitem:` Changed `\c@enumiv` to
`\value of \@listctr` 472
- 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.) 355
- 1992-01-10 ltthm.dtx LaTeX2.09
`\@othm:` (RmS) Check for existence
of theorem environment 436
- 1992-01-14 ltbibl.dtx LaTeX2.09
`\@biblabel:` removed `\hfill` . . . 474
- 1992-01-14 ltsect.dtx 0.0
`\@starttoc:` (RmS) added
`\immediate` to `\openout` as all
`\write` commands are also
executed `\immediate` 446
- 1992-02-26 ltbibl.dtx LaTeX2.09
`\@lbibitem:` Added `\hfill` to
restore left-alignment of
bibliography labels in alpha
style 472
- 1992-03-18 ltdefs.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` 463
`\end@float:` (RmS) changed
`\@esphack` to `\@Espack` 457
- 1992-03-18 ltlists.dtx 0.0
General: RmS: added
`\@nmbrlistfalse` 370
- 1992-03-18 ltmiscen.dtx LaTeX2.09
`\begin:` Changed `\@ignoretrue` to
`\@ignorefalse` (as
documented) 338
- 1992-03-21 ltffsini.dtx v1.2d
General: Renamed `\text` to
`\nfss@text` to make it
internal. 247
- 1992-05-12 ltffsbas.dtx v1.3c
`\extract@alph@from@version:`
Macro added. 173
`\select@group:` Added call to `\ex-`
`tract@alph@from@version.` . . . 173
- 1992-07-26 ltffsbas.dtx v1.9a
`\curr@fontshape:` 165
`\DeclareFontShape:` Introduced
`\DeclareFontShape` 155

<code>\define@newfont:</code>	165	1992-09-18 ltlists.dtx LaTeX2.09
<code>\math@fonts:</code>	172	<code>\item:</code> (RmS) Added warning if
<code>\select@group:</code>	172, 173	<code>\item</code> is used in math mode
<code>\split@name:</code> Added splitting into		1992-09-18 lttab.dtx LaTeX2.09
<code>\f@encoding.</code>	165	<code>\@array:</code> Changed <code>\par</code> to
<code>\wrong@fontshape:</code>	170	<code>\@empty</code> to avoid starting new
1992-07-26 ltfssstrc.dtx v2.0b		row e.g. after <code>\hline</code>
<code>\s@fct@:</code>	214	1992-09-19 ltfssstrc.dtx v2.0c
<code>\selectfont:</code>	199	<code>\try@simple@size:</code>
<code>\try@simple@size:</code>	208, 209	1992-09-21 ltfssini.dtx v1.4d
<code>\try@size@range:</code>	212	<code>\not@math@alphabet:</code> Macro
<code>\use@mathgroup:</code>	205	defined.
1992-08-14 ltbibl.dtx LaTeX2.09		1992-09-22 ltfssbas.dtx v1.91a
<code>\@citex:</code> added missing argument		General: Introduced <code>\tf@size</code> for
braces around <code>\hbox</code> , found by		math size.
Ed Szynter	472	1992-09-22 ltfssstrc.dtx v2.1a
1992-08-14 ltboxes.dtx LaTeX2.09		<code>\getanddefine@fonts:</code> Introduced
<code>\endminipage:</code> (RmS) replaced		<code>\tf@size</code> for math size.
<code>\vskip-\lastskip</code> by <code>\unskip</code>		1992-11-13 ltfssini.dtx v?
(proposed by FMi)	385	<code>\hexnumber@:</code> Made expandable.
1992-08-17 ltbibl.dtx LaTeX2.09		1992-11-23 ltcounts.dtx LaTeX2.09
<code>\@citex:</code> simplified code for		<code>\stepcounter:</code> Replaced <code>{}</code> in
removing leading blanks in		<code>\stepcounter</code> by <code>\begingroup</code>
citation key (proposed by		<code>\endgroup</code> to avoid adding an
Frank Jensen and Kresten		empty ord in math mode
Krab Thorup)	472	1992-11-26 ltboxes.dtx LaTeX2.09
1992-08-19 ltsect.dtx 0.0		<code>\@mpfootnotetext:</code> (RmS) added
<code>\@xsect:</code> (RmS) corrected bug:		protection for <code>\edef</code>
stretch and shrink in argument		1992-11-26 ltfloating.dtx LaTeX2.09
to <code>\hskip</code> previously not		<code>\@footnotetext:</code> (RmS) added
negated	443	protection for <code>\edef</code>
1992-08-19 lthm.dtx LaTeX2.09		<code>\footnote:</code> (RmS) Changed all to
<code>\@othm:</code> (RmS) Changed error		<code>'def'protect'noexpand'protect'noexpand</code>
message to complain about	
undefined counter	436	1992-12-03 ltfssini.dtx v?
1992-08-20 ltfssini.dtx v1.4b		<code>\hexnumber@:</code> Make it accept
<code>\@setsize:</code> Added <code>\@currsize.</code>	260	counters.
1992-08-24 ltdefs.dtx LaTeX2.09		1993-03-08 preload.dtx v2.0b
<code>\@ifnextchar:</code> (Rms)		General: Added 12pt preloads
<code>\@ifnextchar</code> didn't work if its		1993-03-18 ltfssbas.dtx v2.0c
first argument was an equal		General: Changed all <code>\@tempdima</code>
sign.	50	in <code>\@tempdimb</code> to avoid killing
1992-08-24 ltmiscen.dtx LaTeX2.09		<code>\numberline</code>
<code>\begin:</code> Added code to <code>\begin</code> to		1993-03-18 ltfssstrc.dtx v2.1b
remember line number. Used		General: Changed all <code>\@tempdima</code>
by <code>\@badend</code> to display position		in <code>\@tempdimb</code> to avoid killing
of non-matching <code>\begin</code>	338	<code>\numberline</code>
<code>\verb:</code> Changed <code>\verb</code> and		Changed all <code>\@tempdimb</code> in
<code>\@sverb</code> to work correctly in		<code>\@tempdimx</code> to avoid killing
math mode	346	<code>\numberline</code>
1992-08-25 ltsect.dtx LaTeX2.09		1993-03-18 ltfssstrc.dtx v2.1c
<code>\@sect:</code> (FMi) replaced explicit		<code>\DeclareSizeFunction:</code> Added all
setting of <code>\@svsec</code> by call to		args to avoid blanks problems
<code>\@secCNTformat</code>	442	

- 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 288
- 1993-05-16 ltfssbas.dtx v2.0e
`\showwhyphens`: Use `\reset@font` 175
- 1993-07-16 ltfsstrc.dtx v2.1h
 General: Changed layout of info messages 195
- 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.` . . 334
- 1993-08-05 ltpictur.dtx LaTeX2.09
`\circle`: (RMS) Added error message if `\circle` is used in math mode. 429
- 1993-08-05 ltsect.dtx LaTeX2.09
`\@sect`: (RmS) Made sure that `\protect` works correctly in expansion of `\the counter` . . 442
- 1993-08-05 ltspace.dtx LaTeX2e
`\@hspace`: (RmS) Removed superfluous `\leavevmode` in `\@hspace` and `\@hspacer`, as suggested by CAR. 85
- 1993-08-05 lttab.dtx latex2e
`\tabular*`: Replaced `\expandafter\def` by `\@namedef.` 400
- 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. 472
- 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. 285
- 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.` 230
- 1993-09-02 ltfsstrc.dtx v2.1i
 General: Corrected name of `sgen size function.` 195
- 1993-09-03 ltmiscen.dtx LaTeX2.09
`\verbatim@nolig@list`: Replaced `\@noligs` by extensible list . 347
- 1993-09-07 ltmiscen.dtx LaTeX2.09
`\verb@balance@group`: (RmS) Changed definition of `\verb` so that it detects a missing second delimiter. 346
- 1993-09-08 ltmiscen.dtx LaTeX2.09
`\enddocument`: Added warning in case of undefined references. 334
- 1993-09-15 ltfssbas.dtx v2.0g
`\DeclareFontEncoding`: Corrected: `\default@T` to `\default@M.` . 157
- 1993-09-15 ltfsstrc.dtx v2.1j
 General: Corrected spelling of `\noexpand.` 195
- 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. 356
- 1993-11-03 ltmath.dtx LaTeX2e
 General: RmS: Initialized `\everycr` to empty 356
- 1993-11-03 ltpictur.dtx LaTeX2.09
 General: (RmS) changed `\halign` to `\ialign` to initialize `\tabskip` and `\everycr` 414
- 1993-11-11 ltfssini.dtx v2.1a
`\normalfont`: Macro added 261
- 1993-11-11 ltfsstrc.dtx v2.2a
 General: Option concept added for LaTeX2e 195
- 1993-11-14 ltclass.dtx v0.2a
`\@current`: Name changed from `\@currentextension` 555
- `\@fileswithoptions`: Moved resetting of `\default@ds,` `\ds@` and `\declaredoptions` here, from the end of `\ProcessOptions.` 565
- `\@reset@options`: macro added . . 567
- `\@AtEndDocument`: Included extension in the generated

macro name for package and class hooks.	567	1993-11-22 ltclass.dtx v0.2f	
<code>\documentstyle</code> : Added		<code>\@fileswithoptions</code> : Made the default [] not	
<code>\RequirePackage</code>		<code>[\@unknownversion]</code>	564
<code>\@unusedoptionlist</code> stuff.	562	Made the initial version [] not	
<code>\g@addto@macro</code> : Made global	567	<code>[\@unknownversion]</code>	565
<code>\NeedsTeXFormat</code> : made more robust for alternative syntax for other formats.	563	<code>\@ifclasslater</code> : Added //00 so parsing never produces a runaway argument.	556
<code>\ProcessOptions*</code> : Optimise 'empty option' code.	560	General: <code>\@unknownversion</code> removed	550
Stop adding the global option list inside class files.	560	1993-11-22 ltdefns.dtx LaTeX2e	
1993-11-15 ltclass.dtx v0.2b		<code>\@minus</code> : Macro added	36
<code>\documentstyle</code> : Modified to match <code>\ProcessOption*</code>	562	<code>\@plus</code> : Macro added	36
<code>\ProcessOptions*</code> : Star form added.	560	<code>\CheckCommand</code> : Macro added	43
1993-11-17 ltclass.dtx v0.2c		<code>\providecommand</code> : Macro added	43
<code>\@@fileswith@pti@ns</code> : Macro added	567	1993-11-22 lterror.dtx LaTeX2e	
<code>\@badrequireerror</code> : Macro added	568	<code>\c@errorcontextlines</code> : Macro added	64
<code>\@fileswithoptions</code> : Added trap for two <code>\LoadClass</code> commands.	566	1993-11-22 ltfiles.dtx LaTeX2e	
<code>\@twoloadclasserror</code> : Macro added	568	<code>\listfiles</code> : Removed checking for <code>\@unknownversion</code>	100
<code>\CurrentOption</code> : Name changed from <code>\@curroption</code>	555	1993-11-22 llength.dtx LaTeX2e	
<code>\DeclareOption*</code> : Error checking added	559	<code>\@settodim</code> : Macro added	153
<code>\NeedsTeXFormat</code> : Name changed from <code>\NeedsFormat</code>	563	<code>\@settopoint</code> : Macro added	153
<code>\ProcessOptions*</code> : restoring <code>\@fileswith@pti@ns</code> added.	560	<code>\settodepth</code> : Macro added	153
1993-11-18 ltclass.dtx v0.2d		<code>\settoheight</code> : Macro added	153
<code>\documentstyle</code> : Modified		1993-11-22 ltlogos.dtx LaTeX2e	
<code>\RequirePackage</code> stuff.	562	<code>\LaTeXe</code> : Macro added	87
<code>\ExecuteOptions</code> : Use		1993-11-23 ltclass.dtx v0.2g	
<code>\CurrentOption</code> not		<code>\@use@option</code> : Name changed from <code>\@executeoption</code>	561
<code>\reserved@a</code>	561	General: Various macros now moved to latex.tex.	554
<code>\NeedsTeXFormat</code> : <code>\fmtname</code>		Warnings and errors now directly coded.	554
<code>\fmtversion</code> not <code>\@...</code>	563	1993-11-23 ltdefns.dtx LaTeX2e	
1993-11-21 ltfiles.dtx LaTeX2e		<code>\@argdef</code> : Macro added	39
<code>\@missingfileerror</code> : Stop infinite looping on <code>\@er@ext</code>	99	<code>\@ifundefined</code> : Redefined to remove a trailing <code>\fi</code>	49
1993-11-21 ltmiscen.dtx v0.9a		<code>\@newcommand</code> : Macro added	39
<code>\@verbatim</code> : use <code>\verbatim@font</code> instead of <code>\tt</code>	344	<code>\@newenv</code> : Macro interface changed	42
<code>\verb</code> : Use <code>\verbatim@font</code> instead of <code>\tt</code>	346	<code>\@xargdef</code> : Macro interface changed	39
<code>\verbatim@font</code> : Macro added	344	<code>\@yargdf</code> : Avoid <code>\@?@?</code> token	40
		Macro interface changed	40
		<code>\newcommand</code> : Macro reimplemented and extended	39
		<code>\renewcommand</code> : Macro reimplemented and extended	41
		<code>\renewenvironment</code> : Macro reimplemented and extended	42
		<code>\two@digits</code> : Macro added	36

1993-11-23 ltoutput.dtx v0.1a		\@imakebox: macro modified ...	378
\paperheight: Register added ..	490	\@irsbox: redefined to support	
\paperwidth: Register added ...	490	\height	387
1993-11-23 ltoutput.dtx v0.1c		\@isavebox: color support	380
\@enlargepage: Command added	533	extra group	380
\@kludgeins: Insert added	533	\@isavepicbox: extra group	380
\@makecol: Command changed ..	501	\@makebox: default changed from x	
\@specialoutput: Command		to c	378
changed	495	\@makepicbox: macro modified ..	379
\enlargethispage*: Commands		\@savebox: default c not x	380
added	533	\bm@b: macros added	378
1993-11-24 ltfntcmd.dtx v2.1a		\endlrbox: macro added	380
\maybe@ic@: Use \t@st@ic	293	\fbox: extra group	381
\t@st@ic: Macro added	293	\lrbox: color support	380
1993-11-24 ltfssini.dtx v2.1a		macro added	380
General: Removed \xpt stuff ...	261	\makebox: modified	377
1993-11-24 ltlogos.dtx LaTeX2e		\mbox: extra group	378
\LaTeX: Macro changed	87	\minipage: Redefined to support	
1993-11-28 ltclass.dtx v0.2h		extra optional arguments ...	385
\@twoclasseserror: Macro added	568	\newsavebox: Pass the whole of	
General: Assorted commands now		arg 1 to \@ifdefinable	379
in the kernel removed.	554	\parbox: Redefined to support	
Directory syntax checing moved		extra optional arguments ...	382
to dircheck.dtx	554	\raisebox: redefined to support	
Primitive filenames now		\height	387
terminated by space not		\sbox: color support	380
\relax.	554	extra group	380
\endfilecontents: Don't globally		\set@color: color support	379
allocate a write stream (always		macro added	379
use 15)	568	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. ...	567
parser from dircheck	99	General: \@onlypreamble: Many	
1993-11-29 ltoutput.dtx v1.0b		commands declared.	554
\@makecol: \@makespecialcolbox		Removed obsolete	
added	501	\@documentclass	554
\@makespecialcolbox: Command		1993-12-03 lterror.dtx v1.0b	
added	503	\@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 ...	247
1993-11-30 ltoutput.dtx v1.0c		1993-12-04 ltfiles.dtx v0.9b	
\fl@tracemessage: Commands		\@iinput: Macro reimplemented .	99
added	535	\@input: Macro reimplemented .	99
1993-12-01 fontdef.dtx v2.1a		\IfFileExists@: Macro added ..	97
General: Update for LaTeX2e ...	264	\input: Macro reimplemented ..	98
1993-12-01 ltoutput.dtx v1.0e		\InputIfFileExists: Macro	
\@reinserts: Command added .	502	added	98
1993-12-03 ltboxes.dtx v0.1a		1993-12-05 ltfloat.dtx LaTeX2e	
\@argsbox: macro removed	387	\@dblfloatplacement: Command	
\@begin@tempboxa: macro added	378	changed	459
\@end@tempboxa: macro added ..	378	\@xfloat: Command changed ..	454
\@iirsbox: redefined to support		1993-12-05 ltoutput.dtx v1.0f	
\height	387	\@addtobot: Command changed .	515

<code>\@addtocurcol</code> : Command	1993-12-07 ltclass.dtx v0.2m
changed	516
<code>\@addtodblcol</code> : Command	<code>\@fileswithoptions</code> : Reset
changed	527
<code>\@addtonextcol</code> : Command	<code>\CurrentOption</code> 565
changed	523
<code>\@addtotoporbot</code> : Command	1993-12-07 ltoutenc.dtx 1.1
changed	515
<code>\@boxfpsbit</code> : Command added	538
<code>\@flcheckspace</code> : Command	General: Protected all special
added	540
<code>\@flsetnum</code> : Command added	539
<code>\@flsettextmin</code> : Command	characters with <code>\string</code> 105
added	540
<code>\@flstop</code> : Commands added	536
<code>\@flupdates</code> : Command added	541
<code>\@fpsadddefault</code> : Command	1993-12-07 ltoutenc.dtx v1.1
added	537
<code>\@getfpsbit</code> : Command added	538
<code>\@opcol</code> : Command changed	501
Hook added	501
<code>\@outputpage</code> : Command	General: Made all character
changed	505
<code>\@resethfps</code> : Command added	539
<code>\@setfloattypecounts</code> : Command	numbers decimal. 102
added	537
<code>\@setfpsbit</code> : Command added	538
<code>\@shipoutsetup</code> : Command	Removed a lot of equal signs
added	505
<code>\@startcolumn</code> : Command	and the like. 102
changed	510
<code>\@startdblcolumn</code> : Command	1993-12-08 ltboxes.dtx v0.1b
changed	510
<code>\@testfp</code> : Command added	538
<code>\@textfloatsheight</code> : Commands	<code>\@begin@tempboxa</code> : Extra braces
added	537
<code>\@topnewpage</code> : Commands	for color support (braces
changed	493
<code>\@tryfcolumn</code> : Command	removed from other macros) 378
changed	512
<code>\@writesetup</code> : <code>\@startpagehook</code>	<code>\@irsbox</code> : fix typo 387
added	505
<code>\output</code> : Command changed	495
1993-12-06 ltclass.dtx v0.2k	<code>\@parboxto</code> : <code>\endgraf</code> added due
<code>\ExecuteOptions</code> : Preserve	to extra group in
<code>\CurrentOption</code>	561
1993-12-06 ltoutput.dtx v1.0f	<code>\@begin@tempboxa</code> 383
<code>\@specialoutput</code> : Unboxing of 255	<code>\lrbox</code> : move <code>\@endpefalse</code> out of
added to rescue writes	495
1993-12-06 ltoutput.dtx v1.0g	the inner group 380
<code>\@topnewpage</code> : <code>\@floatplacement</code>	1993-12-08 lftncmd.dtx v2.1b
placement bug fixed	493
1993-12-07 ltclass.dtx v0.2l	General: Macros <code>\rm</code> , <code>\bf</code> and <code>\sf</code>
<code>\ProvidesFile</code> : Macro added	559
	moved to classes.dtx 295
	1993-12-08 ltlists.dtx LaTeX2e
	<code>\@item</code> : use <code>\sbox</code> to support
	colour 373
	1993-12-08 ltspace.dtx LaTeX2e
	<code>\@bsphack</code> : Command
	reimplemented 77
	Command reimplemented; late
	birthday present for Chris 77
	<code>\@vbsphack</code> : Command added 80
	1993-12-09 ltboxes.dtx v0.1c
	<code>\@irsbox</code> : fix another typo 387
	1993-12-09 ltclass.dtx v0.2n
	<code>\documentstyle</code> : input 209
	compatibility file. 562
	1993-12-09 ltfiles.dtx v0.9e
	<code>\document</code> : Hook added 90
	1993-12-09 ltmiscen.dtx v0.9e
	<code>\enddocument</code> : Hook added 334
	1993-12-10 ltoutenc.dtx v1.2
	General: Added source code for
	<code>tlenc.sty</code> 102
	1993-12-11 lftncmd.dtx v3.0a
	General: Complete reworking of all
	text commands, using just one
	creator function 288
	italic correction now put in front
	of penalty before glue 288
	newcommands replaced by defs 288
	newfontswitch command
	corrected and changed 288

\DeclareTextFontCommand: Macro changed	290	\IfFileExists: Removed interactive prompting for current directory syntax	10
\emph: Macro changed	291	\strip@prefix: modified, name changed from \stripmeaning.	5
\fix@penalty: Macro added	294	1993-12-13 ltlists.dtx latex2e	
\maybe@ic: Macro name changed	293	\trivlist: Initialised \@itemlabel	370
\maybe@ic@: Macro and name changed	293	1993-12-13 ltmiscen.dtx v0.9h	
\sw@slant: Macro changed	294	\@noligs: Readded \@noligs	347
\textup: Macros changed	291	\@verbatim: Readded \@noligs	344
1993-12-11 ltmath.dtx v0.9g		Removed optional argument of \item	343
General: Added a group around the first argument of \frac to prevent changes (for example font changes) from modifying the contents of the second argument.	356	center: Removed optional argument of \item	342
1993-12-11 ltoutenc.dtx v1.2a		flushleft: Removed optional argument of \item	342
General: Corrected for t1enc, math.	102	flushright: Removed optional argument of \item	342
1993-12-11 ltsect.dtx LaTeX2e		1993-12-13 ltoutenc.dtx v1.2b	
\author: Added default	438	General: Corrected file name in driver code.	102
\title: Added default	438	1993-12-13 lttab.dtx latex2e	
1993-12-11 ltxref.dtx LaTeX2e		\tabbing: Removed optional argument of \item	395
\@setref: Macro added	331	1993-12-14 ltoutput.dtx v1.0i	
\pageref: Macro reimplemented	331	General: Section added to declare all parameters	546
\ref: Macro reimplemented	331	1993-12-15 ltboxes.dtx v0.1d	
1993-12-12 ltoutput.dtx v1.0h		\@iminipage: Changed default from ‘c’ to ‘s’	385
\@cflb: boxmaxdepth setting moved	509	\@iparbox: Changed default from ‘c’ to ‘s’	383
defs changed to lets	509	\minipage: Changed default from ‘c’ to ‘s’	385
\@cflt: name changed	509	extra space removed.	385
\@docclearpage: defs changed to lets	500	\parbox: Changed default from ‘c’ to ‘s’	382
\@makecol: defs changed to lets	502	1993-12-15 ltclass.dtx v0.2p	
\@resethfps: Warnings added: minimal	539	General: Removed extra ‘.’s from \@warnings	554
\@startdblcolumn: defs changed to lets	511	1993-12-16 ltlogos.dtx LaTeX2e	
\@topnewpage: braces removed	493	\LaTeXe: Extended logo by DPC	87
\@tryfcolumn: defs changed to lets	512	1993-12-16 ltmath.dtx v0.9i	
\fl@tracemessage: Commands changed	535	\@eqnocr: use \refstepcounter instead of shortcut	357
1993-12-13 ltclass.dtx v0.2o		General: use \refstepcounter instead of shortcut	356
General: Removed setting \errorcontextlines (now in latex.tex)	554	1993-12-16 ltmiscen.dtx v0.9i	
\documentstyle: compatibility file now latex209.sty.	562	General: \literal added	347
\usepackage: Fixed error handling	563	1993-12-16 ltpage.dtx LaTeX2e	
1993-12-13 ltdirchk.dtx v0.2a		\mark: Init \mark at begin document	477
General: on the ‘docstrip’ pass, do not check openin path	10		

1993-12-16 ltspace.dtx LaTeX2e	initializing mark until the problem is solved.	477
\@bsphack: Corrected optimisation :-)		77
1993-12-16 lttab.dtx latex2e	1993-12-18 ltoutenc.dtx 1.3b	
\@xhline: Measure from middle of vertical rules	General: Fixed typos with \ProvidesPackage lines. Added the \NeedsTeXFormat line. Added the last argument to \DeclareEncoding. Moved the use of the encodings to after their declaration.	409
1993-12-17 ltclass.dtx v0.2q	Replaced the missing last argument to \DeclareFontEncoding.	105
\@documentclasshook: Macro added	1993-12-18 ltoutenc.dtx 1.3c	554
\@fileswithoptions: Add \@compatibility hook	General: Rewrote for the new syntax of \EncodingSpecific.	564
\documentstyle: Match Alan's new code.	Split \EncodingSpecificAccent up into \EncodingSpecific and \DeclareAccent.	562
1993-12-17 ltoutenc.dtx 1.3	1993-12-18 ltoutenc.dtx v1.3a	
General: Added this section	General: Replaced OT3 by XXX	106
Removed all the hackery for use in \DeclareFontEncoding, and redid everything using \DeclareTextFoo.	1993-12-18 ltoutenc.dtx v1.3b	118, 120
Removed the catcode hackery, since the file is only read as a package in the preamble, and removed all the messages on the screen, which just confuse users. Replaced them by the appropriate \ProvidesPackage commands. Added XXXenc.	General: Corrected typos.	105
1993-12-17 ltoutenc.dtx v1.3	Replaced the missing last argument to \DeclareFontEncoding.	102
General: Added	1993-12-18 ltoutenc.dtx v1.3c	
\EncodingSpecificAccent, \EncodingSpecificAccent-edLetter and \EncodingSpecificCommand.	General: A new syntax, separating accent-definitions from encoding-specific definitions, and allowing encoding-specific \chardef, \let, etc.	102
Made Rokicki's encoding a proper encoding scheme rather than a variant of OT1.	Rewrote for the new syntax of \EncodingSpecific.	102
1993-12-17 ltoutput.dtx v1.0j	1993-12-18 ltoutenc.dtx v1.3d	
\@opcol: Hook removed	General: Some T1 stuff had drifted into the OT1 file.	501
\@specialoutput: Page room test added	1993-12-18 ltpage.dtx LaTeX2e	501
\@topnewpage: check for vsize too small added	\@sloppy: Added \emergencystretch	496
Page room test added	1993-12-19 ltclass.dtx v0.2r	493
\@writsetup: —and then removed	\endfilecontents: Different message when ignoring a file	495
\@fl@tracemessage: tracefloatvals made a document command	1993-12-19 ltfntcmd.dtx v3.0b	505
1993-12-17 ltpage.dtx LaTeX2e	General: \@pdef command added	535
\@mark: Removed init \@mark at begin document, since it doesn't work.	Added by ASAJ.	477
\@rightmark: Stopgap solution to mark \leftmark and \rightmark work without	Made \@newfontswitch produce an error if command already exists, and added \@renewfontswitch, ASAJ	
	Other tidying	288
	Some more tidying done	288

Untidying added, so this is now a TEMPORARY version. . .	288	<code>\math@version</code> : New math font setup	164
Wording changes by CAR. . .	296	1994-01-17 <code>ltfssini.dtx v2.1e</code>	
<code>\DeclareOldFontCommand</code> : Corrected and tidied	295	<code>\not@math@alphabet</code> : Message changed	259
<code>\DeclareTextFontCommand</code> : Corrected and tidied	290	1994-01-17 <code>lftsstrc.dtx v2.3a</code>	
1993-12-19 <code>ltspace.dtx LaTeX2e</code>		General: New math font setup . .	195
<code>\@bsphack</code> : There seem to be problems with selfmade birthday presents	78	<code>\check@mathfonts</code> : New math font setup	204
1993-12-20 <code>ltdefns.dtx LaTeX2e</code>		<code>\glb@currsize</code> : New math font setup	201
<code>\@reargdef</code> : Kept old version of <code>\@reargdef</code> , for <code>array.sty</code>	41	<code>\restglb@settings</code> : New math font setup	204
1993-12-20 <code>ltfiles.dtx v0.9m</code>		1994-01-18 <code>ltbibl.dtx LaTeX2e</code>	
<code>\@obsoletefile</code> : Added this command, removed <code>@oldfilewarning</code>	100	<code>\bibliography</code> : Use <code>\@input@</code> so include files are listed.	473
1994-01-05 <code>fontdef.dtx v2.1d</code>		1994-01-18 <code>ltxclass.dtx v0.2t</code>	
General: Removed <code>nf</code> prefix from file names.	266	<code>\@ifclassloaded</code> : Fix typo <code>\@pkgetension</code>	555
1994-01-13 <code>ltmath.dtx v0.9o</code>		1994-01-18 <code>ltfiles.dtx v0.9p</code>	
<code>\@eqnocr</code> : correcting 0.9i	357	<code>\@iffileonpath</code> : Macro added . .	97
General: correcting 0.9i	356	<code>\@input</code> : do not use a different definition for <code>\input@path</code> . . .	99
1994-01-14 <code>ltdirchk.dtx v0.2d</code>		<code>\@input@</code> : Macro added	99
<code>\IfFileExists</code> : Close the <code>texsys.aux</code> output stream . . .	10	<code>\IfFileExists@</code> : New Definition	97
1994-01-15 <code>ltfiles.dtx v0.9o</code>		<code>\include</code> : Use <code>\@input@</code> so include files are listed.	94
<code>\document</code> : move <code>\@preamblecmds</code> after document hook	92	<code>\InputIfFileExists</code> : New Definition	98
1994-01-17 <code>ltxclass.dtx v0.2s</code>		1994-01-18 <code>ltfssini.dtx v2.1f</code>	
<code>\@fileswithoptions</code> : Modify to reduce parameter stack usage	564, 566	<code>\not@math@alphabet</code> : Message corrected	259
General: Added many more <code>\@onlypreamble</code> commands .	554	1994-01-18 <code>ltmiscen.dtx v0.9p</code>	
Wrapped long lines to column 72	554	<code>\@verbatim</code> : Add <code>\global\@inlabelfalse</code> . . .	343
1994-01-17 <code>ltfiles.dtx LaTeX2e</code>		Only add <code>\penalty</code> if in <code>hmode</code>	343
<code>\listfiles</code> : New Version, adds <code>'tex'</code> if needed, and lines up columns	100	1994-01-19 <code>fontdef.dtx v2.1e</code>	
1994-01-17 <code>ltfssbas.dtx v2.1a</code>		General: Added missing setting for symbols in bold version. . . .	271
General: New math font setup . .	154	1994-01-19 <code>ltdirchk.dtx v0.2e</code>	
<code>\curr@math@size</code> : New math font setup	165	<code>\IfFileExists</code> : name changed from <code>\test</code>	9
<code>\everydisplay</code> : New math font setup	164	<code>\input@path</code> : No longer check that an empty group is in the path	11
<code>\everymath</code> : New math font setup	164	<code>\strip@prefix</code> : name changed from <code>\strip@meaning</code> , to match NFSS.	5
<code>\frozen@everydisplay</code> : New math font setup	164	1994-01-19 <code>ltmath.dtx v1.0n</code> classes	
<code>\frozen@everymath</code> : New math font setup	164	<code>\mathindent</code> : Deferred setting of <code>\mathindent</code>	358
		1994-01-20 <code>ltdirchk.dtx v0.2f</code>	
		General: <code>\@copytexsys</code> and the <code>texsys.new</code> file removed	9
		Modify all of <code>ltxcheck</code>	14

\IfFileExists: \@copytexsys removed	10	1994-01-31 ltfntcmd.dtx v3.1b	
1994-01-21 ltclass.dtx v0.2u		General: \@normalsize no longer defined	288
\documentstyle: compatibility file now latex209.def.	562	1994-02-01 ltpage.dtx LaTeX2e	
1994-01-21 ltdirchk.dtx v0.2g		\pagestyle: (DPC) Modify to get nicer error message	475
General: Improve documentation, reorganise docstrip module . . .	1	\thispagestyle: (DPC) Modify to get nicer error message	476
\filename@parse: Minor changes, and add Mac version (:).	11	1994-02-02 ltclass.dtx v0.2x	
\today: Name changed from \stamp, to save memory	9	\@fileswithoptions: Only run the hook and options check if the file was loaded.	566
1994-01-21 ltfloat.dtx LaTeX2e		1994-02-03 ltoutput.dtx v1.0k	
\@xfloat: Added missing percent characters.	454	\@makespecialcolbox: correct mistakes in the documentation	504
1994-01-21 ltmiscen.dtx v0.9s		1994-02-07 ltclass.dtx v0.2y	
\verbatim@font: Removed unnecessary category code hackery.	344	\@fileswithoptions: Run \@compatibility on the first class to start (not the first to finish)	564
1994-01-24 ltdirchk.dtx v0.2h		\@ifclasswith: Add extra ,s so ‘two’ is not matched with ‘twocolumn’	557
\IfFileExists: Stop testing once texsys.aux has been found . . .	10	\ProcessOptions*: Add extra ,s so ‘two’ is not matched with ‘twocolumn’	560
1994-01-24 ltpage.dtx LaTeX2e		1994-02-07 ltfssbas.dtx v2.1c	
\pagestyle: (DPC) Complain if pagestyle is undefined.	475	\DeclareFontEncoding: revert catcode settings earlier	157
1994-01-25 ltdirchk.dtx v0.2i		\DeclareFontShape@: revert catcode settings earlier	155
General: Protect against looping on \@input and \@end.	2	1994-02-08 ltoutput.dtx v1.0k	
1994-01-25 ltfssbas.dtx v2.1b		\@makespecialcolbox: boxmaxdepth setting added	503
\math@version: Corrections for math setup	164	boxmaxdepth setting removed	503
1994-01-25 ltmath.dtx LaTeX2e		General: Documentation and tasks tidied.	478
\bordermatrix: Removed		1994-02-10 ltclass.dtx v0.2z	
\p@renwd.	352	\@documentclasshook: Changed the name from \@compatibility to \@documentclasshook, and added the check for whether \@normalsize has been defined. ASAJ.	554
1994-01-26 ltfsssrc.dtx v2.3c		\@fileswithoptions: Renamed \@compatibility to \@documentclasshook. ASAJ.	564
\check@mathfonts: Correct trace info placement	204	1994-02-10 ltfssbas.dtx v2.1d	
\restgbl@settings: Correct trace info placement	204	\addto@hook: Made \addto@hook long.	177
1994-01-27 ltfntcmd.dtx v3.1a			
\nocorrlist: Only ., used as default for cm fonts	294		
1994-01-29 ltclass.dtx v0.2v			
\@unprocessedoptions: Macro added.	568		
\@fileswithoptions: All options raise error if no \ProcessOptions appears . .	566		
1994-01-31 ltclass.dtx v0.2w			
\g@addto@macro: Use toks register to avoid ‘hash’ problems . . .	567		
1994-01-31 ltfiles.dtx v0.9t			
\document: set \@normalsize or \@normalsize if necessary	91		

1994-02-10 ltffscmp.dtx v2.1d	Long lines wrapped to 72 columns	88
\scan@fontshape: scan away stuff after pt		219
1994-02-22 ltffssini.dtx v2.1g	General: Correct error message . .	262
1994-02-24 ltffsbas.dtx v2.1e	\DeclareFontShape: Separate restoration of catcodes for fd cmds	155
\define@newfont: Separate restoration of catcodes for fd cmds		166
\nfss@catcodes: Separate restoration of catcodes for fd cmds		166
1994-02-25 ltdirchk.dtx v0.2j	General: Remove need for drv file .	1
1994-03-01 ltdirchk.dtx v0.2k	General: Add unstripped module, so that dircheck.dtx may be used with initex	1
1994-03-02 ltboxes.dtx v0.1e	General: Add 2kernel module . .	377
Remove need for drv file		377
1994-03-02 ltclass.dtx v0.3a	General: Remove need for driver file	554
1994-03-03 ltboxes.dtx v0.1f	\@irsbox: Replaced a missing \else	387
1994-03-04 ltfloat.dtx v1.0a	General: Initial version, split from latex.dtx	450
1994-03-04 ltsect.dtx v1.0a	General: Initial version, split from latex.dtx	438
1994-03-04 lttab.dtx v1.0a	General: Initial version, split from latex.dtx	389
1994-03-04 ltvers.dtx v1.0a	General: Initial version, split from latex.dtx	33
1994-03-07 ltboxes.dtx v0.1a	\@mpfootnotetext: Extra group for colour	386
1994-03-07 ltboxes.dtx v1.0a	General: Unify format with other Kernel files	377
1994-03-07 ltdefns.dtx v1.0a	\@italiccorr: Macro added . . .	36
1994-03-07 ltfiles.dtx v1.0a	General: Initial version, split from latex.dtx	88
1994-03-07 ltfinal.dtx v0.1a	General: Add code from the old dump.dtx	624
Initial version, split from latex.dtx		612
move code here from lhyphen.dtx		617
Remove oldcomments environment		612
use \InputIfFileExists not \IfFileExists		617
1994-03-07 ltfloat.dtx v1.0a	\@endfloatbox: (DPC) Extra group for colour	459
\@footnotetext: (DPC) Extra group for colour		467
\@xfloat: (DPC) Extra group for colour		455
1994-03-07 lthyphen.dtx v0.1c	General: move the 2kernel code to ltfinal.dtx	583
1994-03-07 ltlength.dtx v1.0a	\@settodim: (DPC) Extra group for colour	153
1994-03-07 ltlists.dtx v1.0a	General: Initial version, split from latex.dtx	362
Long lines wrapped to 72 columns		362
1994-03-07 ltpage.dtx v1.0a	General: Initial version, split from ltherest.dtx	475
1994-03-07 ltpictur.dtx v0.1a	General: Initial version, split from latex.dtx	411
Long lines wrapped to 72 columns		411
1994-03-07 ltsect.dtx v1.0a	\@changfrom: (DPC)Extra groups for colour	445
1994-03-07 lttab.dtx v1.0a	General: Long lines wrapped to 72 columns	389
1994-03-08 ltclass.dtx v0.3b	General: Modify driver code into ‘new style’	554
1994-03-08 ltdirchk.dtx v1.0a	General: Reorganise driver module into ‘new style’	1
1994-03-08 ltplain.dtx v1.0a	General: Remove need for a driver file.	15

1994-03-10 ltffsbas.dtx v2.2f	1994-03-13 ltfiles.dtx v0.3b
\backslash math@egroup: Changed	\backslash InputIfFileExists: Use new
\backslash begingroup/ \backslash endgroup to	cmd \backslash @addtofilelist 98
\backslash bggroup/ \backslash egroup. 174	1994-03-13 ltffsbas.dtx v2.1g
1994-03-11 ltffsdcl.dtx v2.1b	General: add 2ekernel module to
\backslash DeclareSymbolFontAlphabet@:	omit repeated code 154
Added check against use of	1994-03-13 ltffsdcl.dtx v2.1c
alphabet switch outside of	General: add 2ekernel module to
math mode. 246	omit repeated code 222
\backslash SetMathAlphabet@: Changed	1994-03-14 ltboxes.dtx v1.0b
parameter template in	\backslash @isavebox: Use
temporary macro to catch	\backslash color@setgroup 380
check add below. 235	\backslash @isavepicbox: Use
1994-03-12 ltclass.dtx v0.3c	\backslash color@setgroup 380
\backslash @fileswithoptions: Do not use	\backslash color@begingroup: macro added
\backslash @pr@videpackage to avoid	for colour support 379
timeout 566	\backslash color@endgroup: macro added for
General: Change name from	colour support 379
docclass to ltclass 554	\backslash lrbox: Use \backslash color@setgroup .. 380
\backslash ProvidesFile: Add \backslash wlog 559	\backslash sbox: Use \backslash color@setgroup . . . 380
\backslash ProvidesPackage: Add \backslash wlog .. 558	1994-03-14 ltfloat.dtx 1.0c
use \backslash @gtempa 558	\backslash @xympar: (DPC) Use
1994-03-12 ltdefs.dtx v1.0b	\backslash color@begingroup 463
\backslash @reargdef: New defn, in terms of	1994-03-14 ltfloat.dtx v1.0c
\backslash @yargdef 41	\backslash @endfloatbox: (DPC) Use
\backslash @yargd@f: Name changed from	\backslash color@endgroup 459
\backslash XXX@argdef 40	\backslash @footnotetext: (DPC) Use
1994-03-12 ltdirchk.dtx v1.0b	\backslash color@begingroup, add
General: Change name from	\backslash endgraf 467
dircheck.dtx 1	\backslash @savemarbox: (DPC) Use
Minor edits to the timeouts in	\backslash color@begingroup 462
ltxcheck 1	\backslash @xfloat: (DPC) Use
1994-03-12 ltfloat.dtx v1.0b	\backslash color@begingroup 455
\backslash @savemarbox: (DPC) Extra group	1994-03-15 ltfiles.dtx LaTeX2e
for colour 462	\backslash @missingfileerror: Quit on x or
\backslash @xympar: (DPC) Extra bggroup for	X just like a real error 99
colour 463	1994-03-15 ltfntcmd.dtx v3.2a
1994-03-12 ltplain.dtx v1.0b	General: Adapted to mass
General: Name changed from	formatting 288
lplain. The end of an era 15	Changed \backslash / to \backslash @@italiccorr 288
1994-03-12 ltplain.dtx v1.0e	Removed \backslash @renewfontswitch . 288
General: Replaced remaining	Removed defs of short-forms and
width, height, depth by L ^A T _E X	all sizes except \backslash normalize . 288
macro names to save tokens. . . 15	1994-03-15 ltoutput.dtx v1.0l
1994-03-13 ltcntrl.dtx v1.0c	\backslash @addtocurcol: Changed
\backslash @tfor: (DPC) Add \backslash @tfor so a	\backslash addvspace to \backslash vskip .. 518, 522
single group is correctly	\backslash @combinedblfloats: Removed
treated. 59	boxmaxdepth setting. 510
1994-03-13 ltfiles.dtx LaTeX2e	\backslash @makecol: \backslash maxdepth changed to
\backslash @addtofilelist: Macro added . 100	\backslash @maxdepth 501
\backslash listfiles: Reset	Removed boxmaxdepth setting. 502
\backslash @addtofilelist at begin	\backslash @makespecialcolbox: Removed
document 101	boxmaxdepth setting. 503

<code>\@topnewpage</code> : Corrected and amended warning message . . .	494	1994-03-28 ltsect.dtx v1.0b	General: Split further from ltherest.dtx	438
Warning added: it should be improved	495	1994-03-28 lttab.dtx v1.0b	General: Improve documentation	389
General: Added some warnings when page gets full of top floats.	478	1994-03-28 ltthm.dtx v1.0a	General: Initial version, split from latex.dtx	434
Driver added and further tidying.	478	1994-03-29 ltcounets.dtx v1.0c	General: Create file from parts of ltmiscen and ltherest.	146
Removed duplicated code and corrected docstrip options. . .	478	1994-03-29 ltlength.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	153
Some boxmaxdepth settings removed.	478	1994-03-29 ltmiscen.dtx v1.0d	General: Remove counter macros to ltcntlen	333
1994-03-16 ltclass.dtx v0.3f		1994-03-29 ltpageno.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	328
General: Add pkgindoc package .	582	1994-03-29 ltxref.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	329
1994-03-16 ltfiles.dtx LaTeX2e		1994-03-31 ltbibl.dtx v1.0a	General: Initial version of ltidxbib.dtx, split from ltherest.dtx	471
<code>\listfiles</code> : Move this code directly into <code>\document</code>	101	1994-03-31 ltidxglo.dtx v1.0a	General: Initial version of ltidxbib.dtx, split from ltherest.dtx	469
1994-03-16 ltfiles.dtx v1.0c		1994-04-09 ltcounets.dtx v1.0d	<code>\@newctr</code> : <code>\@nocnterr</code> now has counter name argument	147
<code>\document</code> : (DPC) directly add file list settings	91	<code>\addtocounter</code> : <code>\@nocnterr</code> now has counter name argument .	147	
1994-03-16 ltmiscen.dtx v1.0b		<code>\setcounter</code> : <code>\@nocnterr</code> now has counter name argument	147	
<code>\@verbatim</code> : Remove <code>\global\@inlabelfalse</code> again.	343	<code>\stepcounter</code> : Use <code>\addtocounter</code> to have name checked	147	
1994-03-28 ltalloc.dtx v1.0d		1994-04-09 ltthm.dtx v1.0b	<code>\@othm</code> : Use standard counter error message (FMi)	436
General: Redefinition of ‘new’ allocations removed.	54	1994-04-11 ltclass.dtx v0.3g	<code>\endfilecontents</code> : Add star form, dont write <code>\endinput</code> at the end of the file.	568
1994-03-28 ltdirchk.dtx v1.0d		<code>\ProvidesFile</code> : Protect against weird catcodes.	559	
General: Improve documentation .	1	1994-04-11 ltfssbas.dtx v2.1h	General: Added <code>\defaultscriptratio</code> and <code>\defaultscriptscriptratio</code> . ASAJ.	154
1994-03-28 lterror.dtx v1.0d				
<code>\@invalidchar</code> : (DPC) Comment out (use <code>catcode15</code> instead) . .	67			
General: Remove test for <code>\inputlineno</code> undefined.	64			
1994-03-28 ltfiles.dtx v1.0d				
<code>\document</code> : (DPC) Use <code>\normalsize</code> not <code>\@normalsize</code>	91			
(DPC) remove <code>\@normalsize</code> check	91			
1994-03-28 ltfloat.dtx v1.0b				
<code>\@caption</code> : Use <code>\normalsize</code> not <code>\@normalsize</code>	453			
General: Split further from ltherest.dtx	450			
1994-03-28 ltlists.dtx v1.0b				
General: Improve documentation	361			
1994-03-28 ltmiscen.dtx v1.0c				
General: Improve Documentation	333			
1994-03-28 ltplain.dtx v1.0c				
<code>\newlanguage</code> : Remove some <code>\outer</code> declarations.	17			

<code>\defaultscriptratio</code> : Macro added	175	<code>\no@alphabet@error</code> : Use std LaTeX error macro	154
<code>\defaultscriptscriptratio</code> : Macro added	175	1994-04-18 <code>lftssdcl.dtx</code> ???	
1994-04-12 <code>ltboxes.dtx</code> v1.0c		<code>\DeclareMathAlphabet</code> : Pass correct arg (2 not 3)	233
General: Remove <code>\@acci</code> , now defined in <code>lplain.dtx</code>	383	1994-04-18 <code>lftssdcl.dtx</code> v2.1d	
Remove <code>\@dischyph</code> , now defined in <code>linit.dtx</code>	383	General: Removed surplus <code>\no@alphabet@error</code> (see <code>fam.dtx</code>)	222
1994-04-12 <code>ltdefns.dtx</code> v1.0g		1994-04-18 <code>lftsstrc.dtx</code> v2.3d	
<code>\@dischyph</code> : Define <code>\@dischyph</code> , was previously in <code>ltboxes.dtx</code>	51	General: Changed to new error/warning scheme	195
1994-04-12 <code>lplain.dtx</code> v1.0d		<code>\font@submax</code> : Changed dimen to macro	212
General: Define <code>\@acci</code>	29	<code>\fontsubfuzz</code> : Changed dimen to macro	212
1994-04-12 <code>ltvers.dtx</code> v1.0b		<code>\subst@size</code> : <code>\font@submax</code> and <code>\fontsubfuzz</code> now macros	213
General: Have version info generated automatically.	33	1994-04-19 <code>ltpage.dtx</code> v1.0b	
1994-04-14 <code>lftntcmd.dtx</code> v3.2b		General: Improve documentation	475
General: Macros renamed to non-private forms, JB	288	1994-04-20 <code>lftntcmd.dtx</code> v3.3a	
<code>\DeclareOldFontCommand</code> : Renamed from <code>\@newfontswitch</code>	295	General: Documentation up-dated	288
1994-04-15 <code>ltboxes.dtx</code> v1.0d		New implementation of <code>\nocorr</code>	288
<code>\@isavebox</code> : Added missing procent character.	380	<code>\check@nocorr@</code> : Macros added	292
1994-04-17 <code>ltcounts.dtx</code> v1.0e		<code>\maybe@ic@</code> : <code>\nocorr</code> etc removed from list of tokens to check, leaving only punctuation characters	293
<code>\@newctr</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	147	1994-04-20 <code>ltmiscen.dtx</code> v1.0e	
<code>\addtocounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	147	<code>\enddocument</code> : Changed logic for producing warning messages	335
<code>\setcounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	147	1994-04-21 <code>ltboxes.dtx</code> v1.0e	
1994-04-17 <code>lterror.dtx</code> v1.0h		<code>\@iiiminipage</code> : Extra <code>\bgroup</code> for colour	385
<code>\@nocounterr</code> : New name for error message, old error message (without arg) kept	65	<code>\@mpfootnotetext</code> : Extra <code>\endgraf</code> for colour	386
1994-04-17 <code>lthm.dtx</code> v1.0c		<code>\endminipage</code> : Extra <code>\egroup</code> for colour	385
<code>\@othm</code> : Use new std counter error message (FMi)	436	1994-04-21 <code>ltfinal.dtx</code> v0.1c	
1994-04-18 <code>ltfinal.dtx</code> v0.1b		General: Added comments, set the catcodes of 128–255.	612
General: Initialise <code>\textheight</code> , <code>\textwidth</code> and page style	614	1994-04-22 <code>lftssini.dtx</code> v2.1g	
1994-04-18 <code>lfloat.dtx</code> v1.0d		<code>\not@math@alphabet</code> : Message changed again	259
<code>\@footnotetext</code> : (DPC) Remove Colour support	467	1994-04-23 <code>ltfinal.dtx</code> v0.1d	
<code>\@savemarbox</code> : (DPC) Remove Colour support	462	General: Check that <code>\font@submax</code> is still zero	612
1994-04-18 <code>lftssbas.dtx</code> v2.1i		1994-04-24 <code>ltoutput.dtx</code> v1.0m	
General: Macro <code>\no@alphabet@help</code> removed again	154	<code>\@resetfps</code> : Number 2 changed to <code>\tw@</code>	539
<code>\calculate@math@sizes</code> : Changed message to log only	174	Warning changed	539

<code>\@specialoutput</code> : Message changed to give more info and ‘top’ removed	496	1994-04-28 ltplain.dtx v1.0g General: Turn off overfull box tracing in log	25
<code>\@topnewpage</code> : Message changed to give more info	495	1994-04-29 ltclass.dtx v1.0a General: Change version number to 1 (no other change)	554
Warning message removed as it will be generated later	494	1994-04-29 ltmiscen.dtx v1.0f <code>\@verbatim</code> : <code>\leavevmode</code> added	343
General: Changed <code>\@normalsize</code> to <code>\normalsize</code>	478	Change to <code>\everypar</code> added	344
Corrected unverbbed commands in documentation.	478	1994-04-29 ltoutenc.dtx 1.4a General: Removed <code>\EncodingSpecific</code> . Renamed all the commands. Added <code>\DeclareTextGlyph</code> and <code>\UndeclareTextCommand</code>	106
Removed some long lines and other aesthetic changes.	478	Removed Rokicki’s OT1 variant encoding. Moved the driver to the top.	105
Warning messages changed/corrected.	478	1994-04-30 ltfntcmd.dtx v3.3b General: Documentation up-dated and tidied	288
1994-04-24 ltpictur.dtx v0.1b General: Removed surplus spaces after <code>\hbox to</code> in several cases	411	Prefix <code>frag@</code> changed to <code>frag</code> in <code>\@protecteddef</code>	288
1994-04-25 ltclass.dtx v0.3h General: Removed spurious extra ‘.’s at the end of error messages	554	Title changed	288
1994-04-25 ltfloat.dtx v1.0e <code>\@largefloatcheck</code> : Changed warning message to give more info	459	Warning changed to info message in <code>\@protecteddef</code>	288
Command added	459	1994-04-30 ltoutput.dtx v1.0n <code>\@activechar@info</code> : <code>\@activechar@warning</code> changed to <code>\@activechar@info</code>	504
General: Changed warning messages	450	<code>\@combinedblfloats</code> : Removed rule in <code>topnewpage</code> case	510
Removed obsolete tracing code	450	<code>\@emptycol</code> : Empty column action added: <code>\@emptycol</code>	493
1994-04-27 ltfsstrc.dtx v2.3e General: Corrected item that was forgotten in last change.	195	<code>\@flsetnum</code> : Rogue space removed	539
1994-04-28 lterror.dtx v1.0j <code>\@inmatherr</code> : Macro added	67	<code>\@specialoutput</code> : Cut-off point changed to <code>2\baselineskip</code>	496
1994-04-28 lterror.dtx v1.1c <code>\@inmatherr</code> : Replaced <code>\noexpand</code> with <code>\protect</code>	67	Empty column action added: <code>\@emptycol</code>	496
1994-04-28 ltfssdcl.dtx v2.1e General: Removed all <code>\uppercase</code> in hex num parsing macros	222	Extra empty column added for <code>twocolumn</code> case	496
1994-04-28 ltlists.dtx v1.0c <code>\item</code> : Replaced <code>\@ltxnomath</code> by <code>\@inmatherr</code>	371	Extra empty column added for <code>twocolumn</code> case (wrong, see below)	496
1994-04-28 ltpictur.dtx v0.1c General: bezier curves added	431	<code>\@topnewpage</code> : Added setting of <code>\col@number</code>	493
<code>\multiput</code> : (DPC) Ignore spaces between <code>)</code> (.	413	Cut-off point changed to <code>3\baselineskip</code>	495
(DPC) Macro added	413	Empty column action added: <code>\@emptycol</code>	495
<code>\picture</code> : (DPC) Ignore spaces before <code>(</code>	412	Message changed for Frank	495

General: <code>\@activechar@warning</code>	Set all the catcodes	612
changed to an info message.	1994-05-02 <code>ltfinal.dtx v0.1f</code>	
Added <code>\col@number</code>	General: Set the catcode of	
Documentation tidied.	<code>control-J</code>	621
Empty column action added.	1994-05-02 <code>ltmiscen.dtx v1.0g</code>	
Fixed bug from <code>\dblfigrule</code>	General: Changed 91 to 1991 and	
with <code>\@topnewpage</code>	moved some bits	333
Full of floats action improved.	1994-05-02 <code>ltoutput.dtx v1.0o</code>	
<code>\col@number</code> : Added	<code>\@resethfps</code> : Code shortened	539
<code>\col@number</code>	General: Code of <code>\@resethfps</code>	
<code>\onecolumn</code> : Added setting of	shortened.	478
<code>\col@number</code>	1994-05-03 <code>ltbibl.dtx v1.0b</code>	
1994-05-01 <code>lterror.dtx v1.0k</code>	<code>\nocite</code> : Make <code>\nocite</code> issue a	
<code>\@latexerr</code> : (CAR) Added draft	warning for an undefined	
<code>\@latexinfo</code>	citation key.	473
1994-05-01 <code>ltoutenc.dtx 1.4a</code>	1994-05-03 <code>ltfinal.dtx v0.1f</code>	
General: Added the <code>\a</code> command.	General: Set the catcode of	
Added the <code>\SaveAtCatcode</code> and	<code>control-J</code> to be ‘other’, for use	
<code>\RestoreAtCatcode</code>	in messages.	612
commands.	1994-05-03 <code>lfloat.dtx v1.0f</code>	
Removed the <code>uc/lc</code> table	General: (CAR) Added	
settings, since the T1 <code>uc/lc</code>	<code>\@largefloatcheck</code>	450
table is now the default.	Removed unnecessary braces	
Rewrote for the new	from arguments of	
syntax.	<code>\@ifnextchar</code>	450
1994-05-01 <code>ltoutenc.dtx v1.4a</code>	<code>\enddblfloat</code> :	
General: Removed Rokicki’s	<code>\@largefloatcheck</code> added	458
encoding.	<code>\endfloat</code> : (CAR) Added	
Renamed the commands,	<code>\@largefloatcheck</code>	457
removed the	1994-05-03 <code>ltfssdcl.dtx v2.1f</code>	
<code>\EncodingSpecific</code> command.	General: Renamed	
Turned all slots into decimal.	<code>\@DeclareMathDelimiter</code> to	
Added <code>\a</code>	<code>\@DeclareMathDelimiter</code>	222
1994-05-02 <code>ltcntrl.dtx v1.0l</code>	1994-05-03 <code>ltlists.dtx v1.0d</code>	
<code>\@break@tfor</code> : Macro added (from	<code>\@item</code> : <code>\hskip</code> changed to <code>\kern</code>	372
<code>ltfiles.dtx</code>)	<code>\item</code> : Removed superfluous	
1994-05-02 <code>ltdefns.dtx v1.1f</code>	braces	371
<code>\renewcommand</code> : Removed surplus	1994-05-03 <code>ltmiscen.dtx v1.0h</code>	
<code>\space</code> in error	<code>\@centercr</code> : <code>\@badcrerr</code> replaced	
41	by <code>\@nolnerr</code>	341
<code>\renewenvironment</code> : Removed	1994-05-03 <code>lftab.dtx v1.0d</code>	
surplus <code>\space</code> in error	<code>\@endpbox</code> : Use <code>\@finalstrut</code>	
42	based on depth of	
1994-05-02 <code>ltfiles.dtx v1.0f</code>	<code>\@arstrutbox</code>	410
<code>\@iffilenamepath</code> : <code>\@break@loop</code>	1994-05-04 <code>ltclass.dtx v1.0b</code>	
renamed to <code>\@break@tfor</code>	<code>\NeedsTeXFormat</code> : Changed	
97	wording of the warning	563
<code>\@obsoletefile</code> : Make	1994-05-04 <code>lterror.dtx v1.0m</code>	
<code>\@onlypreamble</code>	<code>\@badcrerr</code> : Error message	
100	removed	67
1994-05-02 <code>ltfinal.dtx v0.1e</code>	1994-05-05 <code>ltbibl.dtx v1.0c</code>	
General: Added setting the ‘letter’	<code>\@citex</code> : Set switch for warning	
catcodes.	and end of run.	472
622		
Added setting the ‘other’		
catcodes.		
622		
Added setting the special		
catcodes.		
621		
Made slot 127 illegal		
622		

- `\nocite`: Do not write page number in `\nocite` warning message. 473
- Set switch for warning and end of run. 473
- 1994-05-05 ltfinal.dtx v0.1g
 - General: Added empty `errhelp`. . . 612
 - `\errhelp`: Set error help empty. . . 625
- 1994-05-05 ltfntcmd.dtx v3.3c
 - `\@math@egroup`: Corrected `\@fontswitch` and added saved versions 295
 - General: Corrected `\@fontswitch` 288
- 1994-05-05 ltmiscen.dtx v1.0i
 - General: Removed braces from `ifnextchar` and `ifstar` arguments 333
- 1994-05-07 lttab.dtx v1.0c
 - `\@maxtab`: Changed `\@firsttab` to `\chardef` 393
 - Changed `\@maxtab` to `\chardef` 393
 - General: Removed definition of `\+` 389
 - Removed surplus braces from `\@ifnextchar` constructs . . . 389
- 1994-05-08 ltfntcmd.dtx v3.3d
 - General: Removed `\@undefinedfonterror` 288
 - `\normalsize`: Removed `\@undefinedfonterror` 296
- 1994-05-09 ltfntcmd.dtx v3.3f
 - General: Replaced all `\next` by `\@let@token` and undo change 3.3e, whatever that was. . . . 288
- 1994-05-10 ltdefs.dtx v1.0n
 - General: (ASAJ) Added `\DeclareProtectedCommand`. . . 36
 - Added `\DeclareProtectedCommand` . . . 44
 - Removed braces around `\@ifundefined` argument. ASAJ. 41
 - `\makeatother`: Added `\makeatletter` and `\makeatother` ASAJ. 51
- 1994-05-10 lterror.dtx v1.0n
 - `\@latexerr`: (ASAJ) Added extra blank lines to `\@latexerr`. . . . 65
- 1994-05-10 ltmiscen.dtx v1.0j
 - `\@sverb`: Slight change in error message text. 345
- 1994-05-11 ltboxes.dtx v1.0f
 - `\@begin@tempboxa`: Use new `\color@setgroup` concept. . . . 378
- `\@iiiminipage`: Use new `\color@setgroup` concept. . . . 385
- `\@mpfootnotetext`: Use new `\color@setgroup` concept. . . . 386
- Use new `\normalcolor` and `\@finalstrut`. 386
- General: Superfluous braces removed from several commands 377
- `\color@setgroup`: macro added for colour support 379
- `\endminipage`: Use new `\color@setgroup` concept. . . . 385
- 1994-05-11 ltclass.dtx v1.0c
 - `\endfilecontents`: Add checks for form feed and tab 568
- 1994-05-11 ltdirchk.dtx v1.0e
 - General: Add `\ProvidesFile` as used in fd files. 4
- 1994-05-11 lterror.dtx v1.0o
 - `\@latexerr`: (ASAJ) Removed one of the extra blank lines to `\@latexerr`. 65
- 1994-05-11 ltlogos.dtx v1.0o
 - `\LaTeX`: Use `\DeclareProtectedCommand`. ASAJ. 87
 - `\LaTeXe`: Use `\DeclareProtectedCommand`. ASAJ. 87
- 1994-05-11 ltoutenc.dtx 1.5a
 - General: Made T1 and OT1 generate packages rather than def files. Renamed the ‘package’ module to ‘teststy’. 105
- 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 ltspc.dtx v1.0o
 - `\,`: Use `\DeclareRobustCommand`. ASAJ. 85
 - `\hspace`: Use `\DeclareRobustCommand`.

ASAJ.	85	1994-05-12 ltfsstrc.dtx v2.3f	
1994-05-12 ltboxes.dtx v1.0g		<code>\selectfont</code> : Use	
<code>\@finalstrut</code> : macro added	388	<code>\DeclareRobustCommand</code>	199
<code>\fbox</code> : New definition, merged with <code>\framebox</code>	381	1994-05-12 ltoutenc.dtx 1.5a	
<code>\framebox</code> : Merged <code>\fbox</code> and <code>\framebox</code>	381	General: Removed the <code>\SaveAtCatcode</code> and <code>\RestoreAtCatcode</code> commands.	118
<code>\normalcolor</code> : macro added for colour support	379	Rewrote for the new syntax.	118, 120
1994-05-12 ltdefs.dtx v1.0p		1994-05-12 ltoutput.dtx v1.0p	
General: (ASAJ) Fixed a bug with <code>\relax</code> which was using <code>\@gobble</code> before defining it. . .	36	<code>\@writsetup</code> : <code>\normalcoloradded</code>	505
Fixed a bug with <code>\relax</code> which was using <code>\@gobble</code> before defining it.	44	General: <code>\normalcoloradded</code> in various places (DPC).	478
1994-05-12 ltfsbas.dtx v2.1j		1994-05-13 ltboxes.dtx v1.0h	
General: New <code>baselinestretch</code> concept	154	<code>\@arrayparboxrestore</code> : New accent system, use <code>\let</code> not <code>\def</code>	384
Replaced hand-protected commands by <code>\DeclareRobustCommand</code> defs	154	1994-05-13 ltcounts.dtx v1.0f	
<code>\f@linespread</code> : New macro	163	General: Removed <code>\@Ialpha</code>	150
<code>\fontencoding</code> : Use <code>\DeclareRobustCommand</code>	161	Removed <code>\@ialpha</code>	150
<code>\fontfamily</code> : Use <code>\DeclareRobustCommand</code>	162	1994-05-13 ltdefs.dtx v1.0q	
<code>\fontseries</code> : Use <code>\DeclareRobustCommand</code>	162	General: (ASAJ) Renamed <code>\DeclareProtectedCommand</code> to <code>\DeclareRobustCommand</code> . Removed <code>\@if@short@command</code>	36
<code>\fontshape</code> : Use <code>\DeclareRobustCommand</code>	162	(ASAJ) Replaces <code>\space</code> by ‘ ’ in <code>\csname</code>	36
<code>\fontsize</code> : Redefined to use <code>\set@fontsize</code>	163	Renamed <code>\DeclareProtectedCommand</code> to <code>\DeclareRobustCommand</code> . Removed <code>\@if@short@command</code> . Moved to after the definition of <code>\@gobble</code>	44
<code>\linespread</code> : New macro	163	1994-05-13 ltdefs.dtx v1.0r	
<code>\mathversion</code> : Use <code>\DeclareRobustCommand</code>	164	General: (ASAJ) Added logging message to <code>\DeclareProtectedCommand</code> . . .	36
1994-05-12 ltfsdcl.dtx v2.1g		Added logging message to <code>\DeclareProtectedCommand</code> . . .	44
General: Allow <code>\relax</code> as undefined command	222	1994-05-13 ltdefs.dtx v1.0s	
Allow <code>\relax</code> 'ed cmds to be declared	222	General: (ASAJ) Added <code>\@backslashchar</code>	36
1994-05-12 ltfsini.dtx v2.1i		(ASAJ) Coded <code>\@ifdefinable</code> more efficiently.	36
General: Moved <code>\fontencoding</code> to fam.dtx	247	Coded more efficiently, thanks to FMi.	41
Moved <code>\fontfamily</code> to fam.dtx	247	1994-05-13 ltfiles.dtx LaTeX2e	
Moved <code>\fontseries</code> to fam.dtx	247	<code>\listfiles</code> : Stop <code>\listfiles</code> being run twice	100
Moved <code>\fontshape</code> to fam.dtx	247		
Moved <code>\fontsize</code> to fam.dtx . .	247		
Moved <code>\mathversion</code> to fam.dtx	247		
Moved <code>\selectfont</code> to tracefnt.dtx	247		

- 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`. 612
- 1994-05-13 ltfinal.dtx v1.0h
 General: Added output enc stuff . 624
- 1994-05-13 ltfloat.dtx v1.0g
 `\@footnotetext`: (DPC) Add new
 style colour support:
 `\normalcolor` 467
 (DPC) Use `\@finalstrut` 467
 `\xfloat`: (DPC) Use
 `\normalcolor` 455
- 1994-05-13 ltfntcmd.dtx v3.3g
 General: Replaced `\@protecteddef`
 by `\DeclareRobustCommand` . 288
- 1994-05-13 ltfssbas.dtx v2.1k
 General: Remove File identification
 ‘typeout’ 154
- 1994-05-13 ltfssbas.dtx v2.1l
 `\DeclareFontEncoding`: Init
 encoding change command . 158
 `\define@newfont`: Use `\@input@`
 for fd files 166
- 1994-05-13 ltfssdcl.dtx v2.1h
 General: Removed file
 identification typeout 222
- 1994-05-13 ltfssini.dtx v2.1j
 General: Removed file
 identification typeout 247
- 1994-05-13 ltfsstrc.dtx v2.3g
 General: Removed typeouts as
 `\ProvidesPackage` writes to
 log. 195
- 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 . . 411
- 1994-05-13 lttab.dtx v1.0d
 `\@contfield`: Colour support . . . 395
 `\@startfield`: Colour support . . 394
 `\@stopfield`: Colour support . . . 395
 `\a`: moved to `ltoutenc` 393
- 1994-05-14 fontdef.dtx v2.1f
 General: Removed .def files. . . . 266
- 1994-05-14 ltfssbas.dtx v2.1m
 `\enc@update`: Macro added 162
- 1994-05-14 ltfssbas.dtx v2.1n
 General: Set defaults for all
 `\f@...` 163
 `\DeclareErrorFont`: Don’t set
 `\f@encoding` 169
 `\DeclareFontEncoding`: Log if
 encoding is redeclared 158
 Only init enc change cmd when
 new encoding 158
- 1994-05-14 ltfssini.dtx v2.1k
 General: Init error font just before
 checking for `fontdef.cfg` 262
 `\reset@font`: Remove surplus
 braces 261
- 1994-05-14 ltfsstrc.dtx v2.3h
 `\selectfont`: Added
 `\enc@update` 200
- 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 143
 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 ltfssbas.dtx v2.1o
 General: encoding cmds changed
 to `enc-cmd` 154
- 1994-05-16 fontdef.dtx v2.1g
 General: Removed
 `\DeclareFontEncoding` for `ot1`
 and `t1` and input .def files
 instead 266
- 1994-05-16 ltalloc.dtx v1.1a
 General: (ASAJ) Split from
 `ltinit.dtx`. 54

1994-05-16 ltcntrl.dtx v1.0a		1994-05-16 ltoutput.dtx v1.0q	
General: (ASAJ) Split from		<code>\@writsetup</code> : Changed setting of	
ltinit.dtx.	56	accents (FMi): with the new	
1994-05-16 ltdefns.dtx v1.1a		encoding setup they can use	
General: (ASAJ) Split from		<code>\let</code> . It could also use the new	
ltinit.dtx.	36	internal commands?	506
1994-05-16 lterror.dtx v1.1a		General: Changed setting of	
General: (ASAJ) Completely new		accents (FMi).	478
error interface.	60	1994-05-16 ltpar.dtx v1.1a	
(ASAJ) Split from ltinit.dtx. . .	60	General: (ASAJ) Split from	
1994-05-16 ltfinal.dtx v1.0i		ltinit.dtx.	69
General: moved output enc stuff to		1994-05-16 ltplain.dtx v1.0h	
lfonts	624	General: Comment out encoding	
1994-05-16 ltfssbas.dtx v2.1p		specific commands	29
<code>\fontsize</code> : Pass <code>\baselinestretch</code>		Remove <code>\@acci</code> and friends	
not <code>\f@linespread</code>	163	again	29
<code>\linespread</code> : Remove surplus		Remove unnecessary def for	
braces	163	<code>\item</code>	28
1994-05-16 ltfssini.dtx v2.1m		<code>\loop</code> : Use Kabelschacht method	27
<code>\@acciii</code> : Define saved versions of		<code>\m@th</code> : Remove unnecessary space	28
accents	263	1994-05-16 ltspace.dtx v1.1a	
1994-05-16 ltlogos.dtx v1.1a		General: (ASAJ) Split from	
General: (ASAJ) Split from		ltinit.dtx.	71
ltinit.dtx.	87	1994-05-17 ltclass.dtx v1.0e	
1994-05-16 ltmath.dtx v1.0k		<code>\@use@ption</code> : Execute option after	
<code>\ensuremath</code> : Use		removing from list, not before	561
<code>\DeclareRobustCommand</code> and		1994-05-17 ltdefns.dtx 1.1b	
add extra braces in math		General: (ASAJ) Added the	
mode	357	<code>\@protect@...</code> commands. . . .	45
1994-05-16 ltoutenc.dtx 1.5h		1994-05-17 ltdefns.dtx v1.1b	
General: <code>\pounds</code> was still using u		General: (ASAJ) Added definitions	
rather than ui shape.	118	for protect.	36
1994-05-16 ltoutenc.dtx v1.5f		(ASAJ) Removed warnings and	
General: enc files now have uc		logging to lterror.dtx.	36
encoding name parts (FMi)	102	Added the discussion of	
Revert code so that the		protected commands, defined	
encoding given is used in		the values that <code>\protect</code>	
<code>\DeclareTextCommand</code> (FMi)	102	should have.	44
1994-05-16 ltoutenc.dtx v1.5g		1994-05-17 ltdefns.dtx v1.1c	
General: Made fontenc.sty use the		General: (ASAJ) Redid definitions	
new mixed-case encoding files. . . .	102	for protect.	36
Removed the lowercasing of the		1994-05-17 lterror.dtx v1.1b	
filename.	143	General: (ASAJ) Moved error stuff	
1994-05-16 ltoutenc.dtx v1.5h		from ltdefns.dtx.	60
General: Added <code>\NG</code> , <code>\ng</code> , <code>\TH</code> , <code>\th</code> ,		1994-05-17 ltfssini.dtx v2.1n	
<code>\DH</code> , <code>\dh</code> , <code>\DJ</code> and <code>\dj</code>	102	<code>\copyright</code> : Really add extra	
Added <code>\r</code> (ring accent) and <code>\k</code>		braces	261
(ogonek) accents.	102	<code>\nfss@text</code> : Added braces to allow	
Fixed a bug with <code>\pounds</code>	102	use in subscripts	261
Removed <code>\P</code> from the OT1		1994-05-17 ltmath.dtx v1.0i	
definitions file.	102	General: Replaced <code>\let</code> by <code>\gdef</code> ,	
1994-05-16 ltoutenc.dtx v1.5i		for indirect definition.	353
General: Fixed a bug with <code>\d</code>	102		

1994-05-17 ltoutenc.dtx v1.5j	1994-05-19 ltplain.dtx v0.1k ltfinal
General: Added braces to <code>\pounds</code>	<code>\showoutput</code> : used <code>\maxdimen</code> not
so it works as a subscript. 102	99999 30
1994-05-18 ltdefs.dtx 1.1c	<code>\showoverfull</code> : used <code>\one</code> not 1 . . . 30
General: (ASAJ) Renamed the	1994-05-19 ltxref.dtx v1.1a
commands, and removed one	General: Extract file ltxref from
which is no longer needed. 45	ltxref.dtx 329
1994-05-18 ltdefs.dtx v1.1c	1994-05-20 ltdefs.dtx v1.1e
General: Redid the discussion and	General: Changed command name
definitions, in line with the	from <code>\@checkcommand</code> to
proposed new setting of	<code>\CheckCommand</code> 36
<code>\protect</code> in the output	<code>\CheckCommand</code> : Changed name
routine. 44	from <code>\@checkcommand</code> to
1994-05-18 ltfinal.dtx v0.1j	<code>\CheckCommand</code> 43
General: Corrected the lcode for	1994-05-20 lterror.dtx v1.1c
d-bar. 612	General: (ASAJ) Added
1994-05-18 ltlogos.dtx v1.1b	<code>\@latex@info@no@line</code> 60
General: (ASAJ) Added the \TeX	(ASAJ) Added missing full
logo. 87	stops. 60
(ASAJ) Made the $\LaTeX 2\epsilon$ logo	(ASAJ) Fixed a bug with
use the text font ‘2’ rather	<code>\@inmatherr</code> 60
than the math font ‘2’. 87	1994-05-20 ltfinal.dtx v0.1l
1994-05-18 ltoutenc.dtx v1.5k	General: Use new font warning
General: Made dotted-i produce	commands 618
‘i’. 102	1994-05-20 ltfloat.dtx v1.0h
Removed braces from <code>\pounds</code>	<code>\@endfloatbox</code> : Restore outer
and <code>\dollar</code> 102	value of <code>@nobreak</code> switch. 459
Replaced <code>\defaultencoding</code>	<code>\outer@nobreak</code> : Macro added:
with <code>\encodingdefault</code> 102	default is to do nothing. 459
1994-05-19 ltbibl.dtx v1.1a	1994-05-20 ltfntcmd.dtx v3.3h
General: Initial version of	General: Use new error commands 288
ltbibl.dtx, split from	1994-05-20 ltfssbas.dtx v2.1q
ltxbib.dtx 471	General: Use new error commands 154
1994-05-19 ltcounts.dtx v1.1a	1994-05-20 ltfssstrc.dtx v2.3i
General: Extracted file from	General: Use new error command
ltxref.dtx 146	names 195
1994-05-19 ltdefs.dtx v1.1d	1994-05-20 ltmiscen.dtx v1.0l
General: (RmS) Added definitions	<code>\@writefile</code> : Added correct
for <code>\@namedef</code> and <code>\@nameuse</code>	setting of <code>\protect</code> 337
again. 36	1994-05-20 ltmiscen.dtx v1.0m
1994-05-19 ltfinal.dtx v0.1k	General: Use new warning
General: Removed <code>\makeat...</code> . . . 612	commands 333
1994-05-19 ltxglo.dtx v1.1a	1994-05-20 ltoutput.dtx v1.0s
General: Initial version of	<code>\@writsetup</code> : Added setting of
ltxglo.dtx, split from	<code>\protect</code> during <code>\shipout</code> 505
ltxbib.dtx 469	General: Added setting of
1994-05-19 ltxlength.dtx v1.1a	<code>\protect</code> during <code>\shipout</code> 478
General: Extract file ltxlength from	1994-05-20 ltxpage.dtx v1.0d
ltxref.dtx 153	<code>\markright</code> : Changed setting for
1994-05-19 ltxpageno.dtx v1.1a	<code>\protect</code> 476
General: Extract file ltxpageno from	1994-05-20 ltxsect.dtx v1.0c
ltxref.dtx 328	General: Correct setting of
	<code>\protect</code> 447

\addcontentsline: Correct setting of \protect.	447	(DPC) New version using long command name.	60
1994-05-21 ltbibl.dtx v1.1b		1994-05-22 ltfloat.dtx v1.0i	
General: Use new warning commands	471	General: Use new warning commands	450
1994-05-21 lterror.dtx v1.1d		1994-05-22 ltoutput.dtx v1.0t	
General: (ASAJ) Made the error commands robust.	60	General: Changed warnings and infos to new commands. . . .	478
1994-05-21 ltfiles.dtx v1.0h		1994-05-22 ltpictur.dtx v0.1e	
General: Use new error commands	88	General: Use new warning cmds .	411
1994-05-21 ltlists.dtx v1.0f		1994-05-23 ltclass.dtx v1.0h	
General: Use new error commands	361	\NeedsTeXFormat: Don't stop completely when format is wrong	563
1994-05-21 ltmiscen.dtx v1.0n		\usepackage: Remove argument if possible	563
1994-05-21 ltsect.dtx v1.0d		1994-05-23 ltdirchk.dtx v1.0f	
General: Use new error commands	438	General: Document \@TeXversion	1
1994-05-21 lttab.dtx v1.0f		1994-05-23 ltfstsrc.dtx v2.3j	
General: Use new error commands	389	General: Removed def of \f@warn@break	212
1994-05-21 ltxref.dtx v1.1b		1994-05-23 ltoutput.dtx v1.0u	
General: Use new warning commands	329	\@activechar@info: Added \MessageBreak	504
\newlabel: Use new warning commands	331	\@writsetup: Changed resetting of \protect after shipout to use \aftergroup	505
1994-05-22 ltclass.dtx v1.0f		General: Added \MessageBreak. .	478
General: Use new warning and error commands	550	Changed resetting of \protect after shipout.	478
1994-05-22 ltdefs.dtx v1.1f		1994-05-24 lterror.dtx v1.2e	
General: Use new warning and error cmds	36	\@latex@info@no@line: Macro added	63
1994-05-22 lterror.dtx v1.1e		1994-05-24 lterror.dtx v1.2f	
General: (ASAJ) Replaced bgroup by begingroup in error messages, to stop extra mathords creeping into math mode.	60	General: (DPC) wrap long lines .	60
1994-05-22 lterror.dtx v1.2a		1994-05-24 ltfntcmd.dtx v3.3i	
General: (ASAJ) Made \GenericError, \GenericWarning and \GenericInfo robust.	60	General: Tidying and typos fixed	288
(ASAJ) Replaced \@generic@message and \@generic@error by \GenericError, \GenericWarning and \GenericInfo.	60	1994-05-24 ltmiscen.dtx v1.0q	
(ASAJ) Replaced \ and tilde by \MessageBreak and \space.	60	\@currenline: Use \@empty as outer default	340
(ASAJ) Replaces \string by \protect in some messages.	60	1994-05-25 ltdirchk.dtx v1.0g	
1994-05-22 lterror.dtx v1.2d		\filename@parse: Mac parser had " typo for :	12
\GenericError: (DPC) Alternative version added for old TeXs	60	1994-05-25 ltfntcmd.dtx v3.3j	
		General: Insertion of \aftergroups to implement \nocorr moved to the end of the group	288
		\check@icr: Macros added	292
		\check@nocorr@: Insertion of \aftergroups moved and defaults set up for efficiency .	292
		\DeclareTextFontCommand: \expandafter inserted	290

Insertion of <code>\aftergroups</code> moved	290	1994-06-09 <code>ltfinal.dtx v1.0n</code> General: For \TeX 2, do not set codes for higher half of character table.	616, 622
1994-05-25 <code>ltoutput.dtx v1.0v</code> General: Extra documentation. . .	478	1994-06-09 <code>ltfntcmd.dtx v3.3k</code> General: Tidying and typos fixed in documentation	288
1994-05-25 <code>ltsect.dtx v1.0e</code> <code>\@dottedtocline</code> : Put braces around argument 4 (the actual toc entry) to avoid font (and possibly other) changes leaking out to the leaders.	448	1994-06-18 <code>ltfntcmd.dtx v3.3l</code> General: Added check for empty text	288
1994-05-25 <code>ltthm.dtx v1.0c</code> General: Modify documentation .	434	1994-06-22 <code>ltfntcmd.dtx v3.3m</code> General: Removed space from <code>\nfss@text</code>	288
1994-05-25 <code>ltvers.dtx v1.0d</code> General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here)	33	Renamed <code>\check@nocorr@</code>	288
1994-05-25 <code>ltxref.dtx v1.1c</code> General: Modify documentation .	329	<code>\check@nocorr@</code> : Renamed <code>\check@nocorr</code> to <code>\text@command</code> to improve <code>\long</code> error message	292
1994-05-26 <code>ltfiles.dtx LaTeX2e</code> <code>\@missingfileerror</code> : Modify message format	99	<code>\DeclareTextFontCommand</code> : Removed space from <code>\nfss@text</code>	290
1994-05-26 <code>ltlogos.dtx v1.1c</code> General: Remove \SLiTeX logo . .	87	1994-06-22 <code>ltmath.dtx v1.2t</code> classes <code>\mathindent</code> : Set <code>\mathindent</code> at the end of the class instead of at begin document	358
1994-05-26 <code>ltmiscen.dtx v1.0r</code> General: <code>\literal</code> removed	347	1994-07-20 <code>ltlogos.dtx v1.1e</code> \LaTeX : Save a few tokens	87
1994-05-26 <code>ltplain.dtx v1.1m</code> <code>\iterate</code> : (CAR) added <code>\long</code> . .	27	\LaTeXe : Save a few tokens	87
<code>\underbar</code> : (CAR/FMi) changed to use box <code>\tw@</code>	28	1994-07-20 <code>ltpage.dtx v1.0h</code> <code>\sloppy</code> : Save a few tokens	477
1994-05-26 <code>ltplain.dtx v1.1p</code> <code>\underbar</code> : (DPC) changed to use <code>\sbox</code>	28	1994-09-16 <code>ltfssbas.dtx v2.1s</code> <code>\nfss@catcodes</code> : Reset [and] as well, just in case	167
1994-05-29 <code>ltfssdcl.dtx v2.1j</code> General: Use new error commands	222	1994-10-07 <code>ltoutenc.dtx v1.5l</code> General: Moved the ogonek accent.	102
1994-05-31 <code>ltfinal.dtx v1.0n</code> General: Renamed <code>lthyphen.*</code> to <code>lthyphen.*</code>	612	1994-10-11 <code>ltdirchk.dtx v1.0h</code> <code>\@TeXversion</code> : Check for \TeX 3.14	13
1994-06-01 <code>ltboxes.dtx v1.0i</code> <code>\@frameb@x</code> : Macro added.	382	General: Modify all of <code>ltxcheck</code> again	14
<code>\@iframebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code>	382	1994-10-12 <code>ltsect.dtx v1.0f</code> General: Doc. typos	438
<code>\fbox</code> : New version, using <code>\@frameb@x</code>	381	1994-10-14 <code>fontdef.dtx v2.2a</code> General: New coding	264
<code>\framebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code>	381	1994-10-14 <code>ltfssini.dtx v2.2a</code> General: New coding for <code>cfg</code> files .	247
1994-06-01 <code>ltlogos.dtx v1.1d</code> \LaTeX : Add <code>\m@th</code> to force math size calculations	87	1994-10-14 <code>ltmiscen.dtx v1.0s</code> General: Move math to other file	333
1994-06-01 <code>ltoutput.dtx v1.0w</code> General: Tidied up typesetting. .	478	1994-10-14 <code>ltplain.dtx v1.1a</code> General: Moved code to other files.	15
1994-06-08 <code>ltfinal.dtx v1.0m</code> General: Add patch file system . .	624		

1994-10-15 ltfssbas.dtx v2.1t	1994-10-24 ltboxes.dtx v1.0k
$\extractalph@from@version$: Warn if math alpha is used outside math 174	\fbox : Inner braces added (to fix latex/1061) 381
1994-10-18 ltboxes.dtx v1.0j	1994-10-25 fontdef.dtx v2.2c
$\@frameb@x$: \leavevmode added 382	General: Added OMSenc.def . . . 266
$\@ifframebox$: \leavevmode moved to $\@frameb@x$ 382	1994-10-25 ltboxes.dtx v1.0l
$\@parboxto$: Macro added to remove misuse of $\@empty$. . 383	$\@isavepicbox$: missing percent (moved from ltpatch) 380
General: stuff from ltpatch done . 377	1994-10-25 ltdefns.dtx v1.2b
\fbox : \long added 381	General: Documentation improvements 36
\mbox : \long added 378	1994-10-25 ltoutenc.dtx 1.6a
\sbox : \long added 380	General: Added \textdollar , \textlbrace , \textrbrace , \textsterling , \textunderline 120
1994-10-18 ltclass.dtx v1.0j	Removed \textlbrace , \textrbrace , \textunderline to give them their proper names. 120
General: Move \listfiles to ltfiles.dtx 550	1994-10-25 ltoutenc.dtx v1.6a
1994-10-18 ltdefns.dtx v1.2a	General: Added \Providetextcommand , \Usetextsymbol , \Usetextaccent , $\Declaretextsymboldefault$, $\Declaretextaccentdefault$, $\Declaretextcommanddefault$, and $\Providetextcommandde-$ $fault$ 102
$\@star@or@long$: macro added . . 38	Added the \Provide commands, and the default definitions. . . 106
General: Add extra test for \endgraf 36	Added the defaults. 114
Add star-forms for all commands 36	Added the files OT1enc.def, T1enc.def and OMSenc.def. . . 114
$\renew@environment$: reset end command 42	Added the OMS encoding. . . . 125
1994-10-18 ltfiles.dtx v1.0i	1994-10-27 ltoutenc.dtx 1.6b
\listfiles : code moved here from ltclass 100	General: Added \textasciicircum \textasciitilde \textbackslash \textbar \textbraceleft \textbraceright \textcompwordmark \textemdash \textendash \textexclamdown \textgreater \textthyphenchar \textthyphen \textless \textquestiondown \textquotedblleft \textquotedblright \textquotedbl \textquoteleft \textquoteright
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 448	
General: Added \normalcolor . . 438	
1994-10-19 ltfssbas.dtx v2.1t	
\DeclareFontEncoding : Add missing \relax 157	
1994-10-23 ltfstrc.dtx v23.k	
$\every@math@size$: Renamed to $\every@math@size$ 202	
1994-10-23 ltmath.dtx v1.0l	
$\@eqnum$: Added \normalcolor since \eqno introduces a subgroup of the displayed math group 355	
\ensuremath : Remove extra braces: but see p 168 of Leslie's book 357	

- `\textunderscore`
- `\textvisiblespace` 120
- Added: `\textemdash`
- `\textendash` `\textexclamdown`
- `\textthyphenchar` `\textthyphen`
- `\textquestiondown`
- `\textquotedblleft`
- `\textquotedblright`
- `\textquoteright` 118
- 1994-10-27 ltoutenc.dtx v1.5d
 - General: Rewrote
 - `\DeclareTextSymbol` to define its argument to use the current encoding by default, to fit with `\DeclareTextCommand`. 106
- 1994-10-27 ltoutenc.dtx v1.6b
 - General: Added `\textbackslash`. 125
 - Added more defaults for OT1. 114
 - Removed the enc.def files 102
 - Removed the files OT1enc.def, T1enc.def and OMSenc.def. . 114
 - Renamed `\textlbrace` to `\textbraceleft` and `\textright` to `\textbraceright`. 125
- 1994-10-29 ltmath.dtx 1.0m
 - General: ASAJ: Added `\DeclareMathOperator`. . . . 348
 - ASAJ: Tidied up documentation. 354
- 1994-10-29 ltmath.dtx v1.0m
 - General: ASAJ: Added `\mathellipsis`, `\mathdollar` and `\mathsterling`. 353
 - ASAJ: Removed `\dag`, `\ddag`. . 353
 - ASAJ: Renamed `\S` and `\P` to `\mathsection` and `\mathparagraph` and made them `\mathchardefs`. 353
- 1994-10-29 ltoutenc.dtx v1.6c
 - General: Added commands like `\dots` for use in text and math. 114
 - Renamed `\P`, `\S`, `\dag` and `\ddag` to `\textparagraph`, `\textsection`, `\textdagger` and `\textdaggerdbl`. 102
- 1994-10-30 ltdefs.dtx v1.2c
 - `\@onelevel@sanitize`: Macro added 51
 - General:
 - (CAR)`\@onelevel@sanitize` added 36
- 1994-10-30 ltdefs.dtx v1.2f
 - General: (DPC)`\newwrite`'s moved to ltfiles 36
- 1994-10-30 ltmath.dtx v1.0n
 - General: ASAJ: Moved the new commands to ltoutenc. 353
- 1994-10-30 ltoutenc.dtx v1.6d
 - General: Added `\DeclareTextCompositeCommand`. 102
 - Added `\textcircled`. 102, 115, 125
 - Added `\t`. 115
 - Added math commands. 102
 - Added OML encoding. . . 102, 115
 - Added the OML encoding. . . . 126
 - Made `\textless` and `\textgreater` come from OML. 115
 - Moved math commands here from ltmath. 118
 - Removed `\textregistered`. . . 115
 - Rewrote `\copyright` to use `\textcircled`. 115
- 1994-10-31 fontdef.dtx v2.2d
 - General: Added OMLenc.def 266
- 1994-10-31 fontdef.dtx v2.2e
 - General: ... and moved further down 266
- 1994-10-31 lfloat.dtx v1.1a
 - `\@dblfloat`: Major changes since two-column and one-column cases merged 454
 - `\@dblflset`: Macro added 453
 - Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved 453
 - `\@endfloatbox`: (DPC/CAR) Extra box added to remove colour resetting from vmode 459
 - `\@floatboxreset`: Macro added . 457
 - `\@footnotetext`: (DPC/CAR) Move colour setting to output routine 467
 - `\@savemarbox`: (DPC/CAR) Extra box added for colour 462
 - `\@setfps`: Macro added 454
 - `\@xdblfloat`: Macros removed: `\@dblft`, `\@xdblfloat` 459
 - `\@xfloat`: (DPC/CAR) Extra box added to remove colour resetting from vmode 455
 - Major changes, removing setting of local variables, space hacks

etc; two-column and one-column cases merged . . .	454	1994-11-04 ltfloat.dtx v1.1b \@footnotetext: (ASAJ) Added	
Reset hook added	455	\protected@edef.	467
\@xympar: (DPC/CAR) Extra box added since needed for floats	463	\footnotemark: Added \protected@xdef to \footnotemark.	468
\fps@dbl: Macro added	454	1994-11-04 ltidxglo.dtx v1.1b \@wrglossary: Added \protected@write to \@wrglossary.	470
1994-10-31 ltoutput.dtx v1.1a \@makecol: (DPC/CAR) Colour resetting moved to here	501	\@wrindex: Added \protected@write to \@wrindex.	470
\@topnewpage: (DPC/CAR) Extra box added to remove colour resetting from vmode	493	General: Removed \if@filesw from \makeindex.	469
(DPC/CAR) Use \color@begingroup for colour	493	\makeglossary: Removed \if@filesw from \makeglossary.	470
(DPC/CAR) Use \normalcolor	493	1994-11-04 ltmiscen.dtx v1.0t \@writefile: Removed setting of \protect. ASAJ.	337
1994-11-02 ltoutenc.dtx v1.6d General: Wrapped lines longer than 70 characters.	102	1994-11-04 ltoutenc.dtx v1.6f General: Added _.	116
1994-11-03 ltclass.dtx v1.0k General: Move \@missingfileerror to ltfiles	554	Added \mathunderscore.	118
1994-11-03 ltdirchk.dtx v1.0i General: Generate an error if latex.ltx not used with clean initex	1	1994-11-04 ltpage.dtx v1.0e \markright: Added \@unexpandable@protect. ASAJ.	476
1994-11-03 ltfiles.dtx v1.0j \@missingfileerror: Move here from ltclass	99	1994-11-04 ltsect.dtx 1.0h \@sect: (ASAJ) Added \protected@edef.	442
1994-11-04 ltboxes.dtx v1.0m \@mpfootnotetext: Added \protected@edef. ASAJ. . .	386	General: (ASAJ) Added \protected@xdef to \thanks.	438
1994-11-04 ltdefs.dtx v1.2e General: Added \set@display@protect to \typeout. ASAJ.	36	1994-11-04 ltsect.dtx v1.0h General: Added \protected@write to \addtocontents. ASAJ. . .	447
Added commands for setting and restoring \protect. ASAJ.	46	\addcontentsline: Added \protected@write to \addcontentsline. ASAJ. . .	447
Rewrote protected short commands using \x@protect. ASAJ.	45	1994-11-04 lttab.dtx v1.0h \@mkpream: (ASAJ) Added \@unexpandable@protect to \@mkpream.	407
1994-11-04 lterror.dtx v1.2g General: Added \set@display@protect to \Generic* commands. ASAJ.	60	\multicolumn: (ASAJ) added \set@typeset@protect. . . .	403
1994-11-04 ltfiles.dtx v1.0k \nofiles: Added setting of \protected@write, \makeindex and \makeglossary to \nofiles. ASAJ.	93	1994-11-04 ltxref.dtx v1.1d \label: (ASAJ)Added \protected@edef	331
\protected@write: Macro added ASAJ.	94	(ASAJ)Added \protected@write	331
		1994-11-05 ltboxes.dtx v1.0n \@mpfootnotetext: Colour resetting for footnotes moved	

to endminipage: as for main page.	386	1994-11-06 ltssini.dtx v2.2b	\@setsize: Use	
\color@endbox: macro added for colour support	379	1994-11-06 ltfssstrc.dtx v2.3k	\@typeset@protect	260
\color@hbox: macro added for colour support	379	\glb@currsiz: New implementation		201
\endminipage: Colour resetting for footnotes moved to here: as for main page.	385	\try@simples: New implementation		212
1994-11-05 ltboxes.dtx v1.0o		\try@size@substitution: New implementation		212
\@mpfootnotetext: Colour groups restored here.	386	\tryis@simple: New implementation		212
1994-11-05 ltfloat.dtx v1.1c		1994-11-07 fontdef.dtx v2.2f	General: (DPC) Add	
\@dblflset: Add compatibility with old version of \@xfloat.	453	\DeclareMathSizes declarations		271
\@endfloatbox: Use new \color@hbox concept.	459	(DPC) Updated to use	\ProvidesFile	266
\@footnotetext: Removed		1994-11-07 ltfiles.dtx v1.0l	\@unused: move here from ltdefns, remove duplicate \@mainaux	90
\normalcolor (again)	467	1994-11-07 ltfiles.dtx v1.0m	\document: Renamed \every@size to \every@math@size.	91
\@savemarbox: Use new \color@hbox concept.	462	1994-11-07 preload.dtx v2.1e	General: (DPC) Updated to use	
\@setfps: Add compatibility with old version of \@xfloat.	454	\ProvidesFile		284
\@xfloat: Add compatibility with old version of \@xfloat: but the arguments, provided at exorbitant cost, are now completely ignored	454	1994-11-09 ltboxes.dtx v1.0p	\@finalstrut: Revert	
Use new \color@hbox concept.	455	\finalstrut: Revert	\finalstrut to 2.09 equivalent (from ltpatch)	388
\@xympar: Use new \color@hbox concept.	463	General: more colour changes.		377
1994-11-05 ltoutenc.dtx v1.6g		1994-11-09 ltfssbas.dtx v2.1v	\@vpt: (DPC) macros added, from setsize.dtx	177
General: Added setting of \@typeset@protect to \patterns and \hyphenation.	113	(DPC) reduce save stack usage latex/1742		177
1994-11-05 ltoutput.dtx v1.1b		1994-11-10 ltbibl.dtx v1.1c	General: Fix \nocite{*}	471
\@topnewpage: Use new \color@hbox concept.	493	\nocite: Fix \nocite{*}		473
\@writsetup: Change protect settings for new-style, protect-free aux-files.	505	1994-11-10 ltmath.dtx v1.2v classes	eqnarray: Added value of	
Use new \color@hbox concept.	505	\parskip to	\abovedisplayskip to compensate for negative	
1994-11-05 ltoutput.dtx v1.1c		\topsep		359
\@beginDvi: Added macro	509	1994-11-10 ltoutput.dtx v1.1e	\@writsetup: Modify \protect setting	505
\@beginDvibox: Added macro	490	1994-11-10 ltplain.dtx v1.1b	General: (CAR) added patch to	
\@writsetup: Add new		\loop.		15
\AtBeginDvi concept	505	\iterate: (CAR) added extra		27
\AtBeginDvi: Added macro	490	\relax		
1994-11-06 ltfssbas.dtx v2.1u				
\cf@encoding: New macro	163			
\DeclareFixedFont: Renamed				
\every@size to				
\every@math@size.	156			

1994-11-11 ltspace.dtx v1.2a		1994-11-17 ltdirchk.dtx v1.0j	
\backslash : (DPC) Make robust	75	General: \backslash @tempa to \backslash reserved@a .	1
1994-11-12 ltfntcmd.dtx v3.3o		1994-11-17 lterror.dtx v1.2h	
\backslash normalsize: Added		General: \backslash @tempa to \backslash reserved@a	60
\backslash MessageBreak	296	1994-11-17 ltfiles.dtx v1.0n	
1994-11-12 ltlists.dtx v1.2b ltspace		General: \backslash @tempa to \backslash reserved@a	88
\backslash endtrivlist: Changed order of		1994-11-17 ltfinal.dtx v1.0o	
tests to make \backslash noitemerror		General: \backslash @tempa to \backslash reserved@a	612
correct: end of an era.	370	1994-11-17 ltfloat.dtx v1.1e	
1994-11-12 ltmiscen.dtx v1.0u		General: \backslash @tempa to \backslash reserved@a	450
center: Changed end macro to		1994-11-17 ltfntcmd.dtx v3.3p	
\backslash def: safer and consistent . .	342	General: \backslash @tempa to \backslash reserved@a	288
flushleft: Changed end macro to		1994-11-17 ltfsbas.dtx v2.1w	
\backslash def: safer and consistent . .	342	General: \backslash @tempa to \backslash reserved@a	154
flushright: Changed end macro		1994-11-17 ltfsdcl.dtx v2.1m	
to \backslash def: safer and consistent	342	General: \backslash @tempa to \backslash reserved@a	222
1994-11-12 ltplain.dtx v1.1c		1994-11-17 ltfsstrc.dtx v2.3l	
General: Comment out more		General: \backslash @tempa to \backslash reserved@a	195
encoding specific commands .	29	1994-11-17 ltmath.dtx v1.0o	
1994-11-12 ltspace.dtx v1.2b		General: \backslash @tempa to \backslash reserved@a	348
\backslash addpenalty: Corrected error		1994-11-17 ltmiscen.dtx v1.0v	
message	81	General: \backslash @tempa to \backslash reserved@a	333
\backslash addvspace: Corrected error		1994-11-17 ltoutenc.dtx v1.6h	
message	81	General: (DPC) \backslash @tempa to	
1994-11-13 ltspace.dtx v1.2c		\backslash reserved@a	102
\backslash addpenalty: Recorrected error		1994-11-17 ltoutput.dtx v1.1h	
message	81	General: \backslash @tempa to \backslash reserved@a.	478
\backslash addvspace: Recorrected error		1994-11-17 ltpictur.dtx v1.0f	
message	81	General: \backslash @tempa to \backslash reserved@a	411
1994-11-14 ltoutput.dtx v1.1f		1994-11-17 ltsect.dtx v1.0i	
\backslash @begindvi: Use normal box		General: \backslash @tempa to \backslash reserved@a	438
register: why a box?	509	1994-11-17 lttab.dtx v1.0j	
\backslash @begindvibox: Use normal box		General: \backslash @tempa to \backslash reserved@a	389
register: why a box?	490	1994-11-18 ltboxes.dtx v1.0r	
\backslash @writsetup: Modify new		\backslash color@vbox: macro added for	
\backslash AtBeginDvi concept	505	colour support	379
General: Removed old definition of		1994-11-18 ltfinal.dtx v1.0n	
\backslash @testfp.	478	General: re-allow slots 127–255 . .	622
1994-11-14 ltspace.dtx v1.2d		1994-11-18 ltfsbas.dtx v2.1x	
\backslash : (DPC) Macro modified	75	General: (DPC) use \backslash reserved@f	
1994-11-14 lttab.dtx v1.0i		not \backslash next	154
\backslash tabularnewline: (DPC) Macro		1994-11-18 ltfsdcl.dtx v2.1m	
added	402	\backslash DeclareMathDelimiter: (DPC)	
1994-11-16 fontdef.dtx v2.2h		\backslash expandafter instead of	
General: (DPC) Removed \backslash { and		\backslash next	239
\backslash }	266	1994-11-18 ltfsstrc.dtx v2.3m	
1994-11-17 ltboxes.dtx v1.0q		General: \backslash next to \backslash reserved@f .	195
General: \backslash @tempa to \backslash reserved@a	377	1994-11-18 ltmath.dtx v1.0p	
1994-11-17 ltclass.dtx v1.0l		\backslash phantom: (DPC) colour support	350
General: \backslash @tempa to \backslash reserved@a	550	(DPC) use \backslash expandafter	
1994-11-17 ltcntrl.dtx v1.0b		instead of \backslash next	350
General: \backslash @tempa to \backslash reserved@a	56	\backslash prime@s: (DPC) use \backslash @let@token	
1994-11-17 ltdefs.dtx v1.0g		instead of \backslash next and	
General: \backslash @tempa to \backslash reserved@a	36	\backslash expandafter instead of \backslash nxt	353

<code>\smash</code> : (DPC) colour support . . .	351	1994-11-30 <code>ltfiles.dtx</code> v1.0o	
(DPC) use <code>\expandafter</code>		<code>\@dofilelist</code> : Macro added	101
instead of <code>\next</code>	351	<code>\listfiles</code> : Use <code>\@dofilelist</code> .	100
1994-11-21 <code>lfloat.dtx</code> v1.1f		<code>\nofiles</code> : There is no	
<code>\@endfloatbox</code> : Added reset of		<code>\@gobblethree</code>	93
minipage flag	459	1994-11-30 <code>ltfssbas.dtx</code> v2.1y	
Corrected position of		<code>\fontshape</code> : Use <code>\@current@cmd</code> in	
<code>\outer@nobreak</code>	459	<code>\@enc@update</code> . ASAJ.	162
<code>\@marginparreset</code> : Macro added	462	1994-11-30 <code>ltmath.dtx</code> 1.0q	
<code>\@savemarbox</code> : Added		General: ASAJ:	
<code>\@setminipage</code> etc	462	<code>\DeclareMathOperator</code> moved	
Added resetting of size and font	462	to <code>AMS\LaTeX</code>	348
Changed to <code>\color@vbox</code>	462	1994-11-30 <code>ltmiscen.dtx</code> v1.0w	
Use <code>\@setnobreak</code> etc	462	<code>\enddocument</code> : (DPC) Do	
<code>\@setminipage</code> : Macro added . . .	457	warnings even for <code>\nofiles</code> .	335
<code>\@setnobreak</code> : Macro added	457	(DPC) Use <code>\@dofilelist</code>	335
<code>\@xfloat</code> : Added <code>\@setminipage</code>	455	1994-11-30 <code>ltoutenc.dtx</code> 1.7a	
Added resetting of size and font	455	General: Redefined <code>\a</code> for the new	
Changed to <code>\color@vbox</code> so		scheme.	114
that large floats overflow at the		1994-11-30 <code>ltoutenc.dtx</code> v1.6g	
bottom	455	General: Removed new definitions	
Missing percents reinserted after		of <code>\patterns</code> and	
4, 8: these are not numbers.	454	<code>\hyphenation</code> , since	
Use <code>\@setnobreak</code>	455	encoding-specific commands	
<code>\@xympar</code> : Changed to		now expand in the mouth. . .	113
<code>\color@vbox</code>	463	1994-11-30 <code>ltoutenc.dtx</code> v1.7a	
1994-11-21 <code>ltoutput.dtx</code> v1.1i		General: Added new code for	
<code>\@addtocurcol</code> : Added		encoding-specific commands.	
<code>\if@nobreak</code> test before float		These now expand in the	
box	518, 522	mouth, which means that	
<code>\@specialoutput</code> : Added		ligaturing and kerning can	
<code>\if@nobreak</code> test	498	happen.	102
<code>\@topnewpage</code> : Changed to		Always load the <code>enc.def</code> file, so	
<code>\color@vbox</code>	493	that the default encoding for	
1994-11-22 <code>ltfssdcl.dtx</code> v2.1o		the commands will change. .	143
General: wrap long lines	222	Redefined <code>\@changed@cmd</code>	
1994-11-22 <code>ltoutenc.dtx</code> v1.6i		to expand in the mouth.	106
General: Corrected <code>\dots</code> so that		Removed <code>\@changed@x@mouth</code>	
there's no kerning in		since <code>\@changed@x</code> now	
monowidth fonts.	102	expands in the mouth.	106
Corrected typo with		Rewrote <code>\@text@composite</code> so it	
<code>\mathunderscore</code>	102	allows an empty argument, or	
Fixed empty accents. Again. . .	102	an argument containing lots of	
1994-11-24 <code>ltdefns.dtx</code> v1.2h		commands.	108
<code>\@newenv</code> : Added test for <code>\endgraf</code>	42	1994-12-01 <code>ltfinal.dtx</code> v1.0p	
1994-11-25 <code>ltplain.dtx</code> v1.1f		General: Renamed <code>lthyphen.*</code> to	
General: (DPC) Comment out lots		<code>hyphen.*</code>	612
of obsolete code	15	1994-12-01 <code>lthyphen.dtx</code> v1.0g	
1994-11-26 <code>lfloat.dtx</code> v1.1b		General: Rename <code>lthyphen.ltx/cfg</code>	
<code>\footnote</code> : (ASAJ) Added		to <code>hyphen.ltx/cfg</code>	583
<code>\protected@xdef</code>	467	1994-12-01 <code>ltplain.dtx</code> v1.1g	
1994-11-28 <code>ltcntrl.dtx</code> v1.0c		General: (DPC) More doc changes	15
General: Documentation			
improvements	56		

1994-12-02 fontdef.dtx v2.2i		1994-12-10 ltfntcmd.dtx v3.3q	
General: Commented out <code>\ldots</code> .		<code>\@@math@egroup</code> : Don't read	
ASAJ.	264	arguments	295
1994-12-02 ltfssini.dtx v2.2c		<code>\check@nocorr@</code> : Use <code>\space</code>	
<code>\copyright</code> : <code>\copyright</code> is now in		command for comparison . . .	292
ltoutenc. ASAJ	261	1994-12-10 ltfssdcl.dtx v2.1p	
1994-12-02 ltlists.dtx v1.0e		<code>\document@select@group</code> :	
<code>\@trivlist</code> : RmS: Added check		Surround with braces (add	
for looping	369	fourth arg)	226
1994-12-02 ltoutenc.dtx 1.7b		<code>\select@group</code> : Surround with	
General: Redefined <code>\a</code> properly. .	114	braces (add fourth arg)	224
1994-12-02 ltoutenc.dtx v1.7b		1994-12-10 ltoutenc.dtx v1.7e	
General: Fixed a bug with <code>\a</code> . . .	102	General: Added documentation for	
1994-12-04 lthyphen.dtx v1.0h		the OML encoding.	102
General: Documentation edits for		Replaced width with <code>\@width</code>	
/1989	583	and ditto height in vrules. . .	102
1994-12-05 ltoutenc.dtx v1.7c		1994-12-14 ltoutenc.dtx v1.7f	
General: Added braces to		General: Added braces to	
<code>\textcircled</code>	102	<code>\copyright</code> so it works	
1994-12-06 ltfssbas.dtx v2.1z		unbraced in subscripts.	102
<code>\DeclareFontEncoding</code> : use		Added check for math mode in	
<code>\nfss@catcodes</code>	157	<code>\@changed@cmd</code>	102
<code>\nfss@catcodes</code> : Added tab char		Commented out	
as well	167	<code>\textasciicircum</code> ,	
1994-12-08 ltoutenc.dtx v1.7d		<code>\textasciitilde</code> ,	
General: Added <code>\null</code> and <code>\sh@ft</code>		<code>\textbackslash</code> , <code>\textbar</code> ,	
to <code>\b</code> and <code>\d</code>	102	<code>\textgreater</code> ,	
1994-12-08 lttab.dtx v1.0k		<code>\textthyphenchar</code> ,	
<code>\@array</code> : Add <code>\tabularnewline</code> .	401	<code>\textthyphen</code> and <code>\textless</code> to	
<code>\tabularnewline</code> : (DPC) Made it		save memory.	102
<code>\relax</code>	402	1995-01-12 ltmath.dtx v1.2y classes	
1994-12-09 ltbibl.dtx v1.1d		<code>\@eqnnum</code> : Added <code>\normalcolor</code> .	358
<code>\bibliographystyle</code> : (DPC)		1995-03-03 ltoutenc.dtx 1.7g	
Allow use in preamble.	473	General: Corrected an error in	
1994-12-10 ltfloat.dtx v1.1g		documentation referring to the	
<code>\@dblfloat</code> : Old version reinstated		tabular rather than the	
temporarily	454	tabbing environment.	114
<code>\@dblflset</code> : Macro removed		1995-04-02 ltfntcmd.dtx v3.3r	
temporarily	453	<code>\@@math@egroup</code> : Read them again	
Old version reinstated		to be able to add <code>\relax</code> . . .	295
temporarily	453	1995-04-02 ltfssdcl.dtx v2.1q	
<code>\@setfps</code> : Macro removed		<code>\document@select@group</code> : fix	
temporarily	454	problem for pr/1275	226
<code>\@xdblfloat</code> : Macros reinserted		<code>\select@group</code> : fix problem for	
temporarily	459	pr/1275	224
<code>\@xfloat</code> : Old version reinstated		<code>\set@mathdelimiter</code> : fix pr/1329	242
temporarily	454	1995-04-02 ltfssini.dtx v2.2d	
Sanitisation added temporarily	454	<code>\not@math@alphabet</code> : add	
General: Some temps reinserted		<code>\noexpand</code> to second part of	
temporarily	450	message	259
<code>\fps@dbl</code> : Macro removed		1995-04-21 ltclass.dtx v1.0m	
temporarily	454	<code>\DeclareOption*</code> : Made long	
		/1498	559

<code>\endfilecontents</code> : Close input check stream: latex/1487 ...	568	1995-04-25 ltsect.dtx v1.0k <code>\@dottedtocline</code> : Added <code>\hbox</code> around dots.	448
1995-04-21 ltfinal.dtx v1.0q General: Allow initial patch level 0	624	1995-04-27 ltboxes.dtx v1.0s <code>\@framebox</code> : Move <code>\leavevmode</code> for graphics/1512	382
1995-04-21 ltoutenc.dtx v1.7h General: Added <code>\null \k</code> latex/1274	102	<code>\@ifframebox</code> : Move <code>\leavevmode</code> for graphics/1512	382
1995-04-22 ltfiles.dtx v1.0p <code>\includeonly</code> : Allow blanks in argument	94	<code>\@iirsbox</code> : Move <code>\leavevmode</code> for graphics/1512	387
1995-04-22 ltmiscen.dtx v1.0x General: Removed extra def of <code>\@gobble</code>	333	<code>\@irsbox</code> : Move <code>\leavevmode</code> for graphics/1512	387
1995-04-23 ltsect.dtx v1.0j <code>\addcontentsline</code> : Use <code>\contentsline</code> internally. ..	447	<code>\fbox</code> : Move <code>\leavevmode</code> for graphics/1512	381
1995-04-24 ltbibl.dtx v1.1e <code>\@citex</code> : Add <code>\mbox</code> to undefined case: latex/1239.	472	<code>\raisebox</code> : Move <code>\leavevmode</code> for graphics/1512	387
1995-04-24 ltbibl.dtx v1.1f <code>\bibcite</code> : Make <code>\@onlypreamble</code> /1388.	472	1995-04-27 ltfiles.dtx v1.0r <code>\document</code> : Added <code>\global</code> to support groups in hook	91
1995-04-24 ltcntrl.dtx v1.0d <code>\@for</code> : Dont expand second argument with <code>\edef</code> : /1317 (DPC)	58	1995-04-27 ltmiscen.dtx v1.0y <code>\enddocument</code> : <code>\@checkend</code> moved after hook	334
1995-04-24 ltoutput.dtx v1.1j <code>\fl@tracemessage</code> : Do not add to kernel unless ‘trace’ specified	535	1995-04-27 ltplain.dtx v1.1i General: Move <code>\hang</code> and <code>\textindent</code> to latex209.def .	28
1995-04-24 ltoutput.dtx v1.1l <code>\@begindivbox</code> : Add <code>\vbox</code> latex/1392	490	1995-04-29 ltcntrl.dtx v1.0e General: Moved init of <code>\protect</code> to ltdefns.dtx	59
<code>\@writsetup</code> : Reset <code>\@</code> latex/1451 (DPC)	506	Removed unused defs for <code>\@setprotect</code> and <code>\@resetprotect</code>	59
1995-04-24 ltpage.dtx v1.0f <code>\fussy</code> : reset <code>\emergencystretch</code> latex/1344	477	1995-04-29 ltdefns.dtx v1.2j <code>\protect</code> : Init <code>\protect</code> here ...	47
1995-04-24 ltplain.dtx v1.1h <code>\newlanguage</code> : Remove remaining <code>\outer</code> declarations.	17	1995-04-29 ltpar.dtx v1.1b General: (TO) Comments clean-up.	69
1995-04-24 ltxref.dtx v1.1e <code>\newlabel</code> : Make <code>\@onlypreamble</code> for /1388.	331	1995-05-02 ltsect.dtx v1.0l <code>\@dottedtocline</code> : Don’t reset to <code>\rmfamily</code>	448
1995-04-25 ltdefns.dtx v1.2i <code>\@check@</code> : Make <code>\long</code> for latex/1346	43	1995-05-03 ltsect.dtx v1.0m General: TO: Promoted documentation to doc.sty standard	438
<code>\new@environment</code> : Parse arguments slowly but safely /1507	41	1995-05-06 ltsect.dtx 1.0n <code>\@secntformat</code> : Use <code>\quad</code> instead of <code>\hskip</code>	444
1995-04-25 ltfiles.dtx v1.0q <code>\document</code> : Removed execution of <code>\every@size</code> latex/1407 ...	91	<code>\@sect</code> : Added <code>\relax</code> after <code>\@secntformat</code> just in case	442
		1995-05-07 ltboxes.dtx v1.0t General: Use <code>\hb@xt@</code>	377
		1995-05-07 ltdefns.dtx v1.2k <code>\hb@xt@</code> : Macro added	36
		1995-05-07 ltmath.dtx v1.0r General: Use <code>\hb@xt@</code>	348

1995-05-07 ltoutput.dtx v1.1m	1995-05-17 ltclass.dtx v1.0o
General: Use <code>\hb@xt@</code> 478	<code>\g@addto@macro</code> : Make long for
1995-05-07 ltpictur.dtx v1.0g	latex/1522 567
General: Use <code>\hb@xt@</code> 411	1995-05-17 ltlists.dtx v1.0g
1995-05-07 ltplain.dtx v1.1j	<code>\@item</code> : Removed surplus braces . . 373
General: Use <code>\hb@xt@</code> 15	<code>\@nbitem</code> : Removed surplus
1995-05-07 ltsect.dtx v1.0o	braces 373
General: Use <code>\hb@xt@</code> 438	<code>enumerate</code> : Use <code>\thr@@</code> and
1995-05-07 lttab.dtx v1.0l	remove surplus braces 374
General: Use <code>\hb@xt@</code> 389	<code>itemize</code> : Use <code>\thr@@</code> 375
1995-05-08 ltbibl.dtx v1.1g	1995-05-18 ltfloat.dtx v1.1j
<code>\@citex</code> : Use <code>\@firstofone</code> 472	<code>\@makefnmark</code> : Added
<code>\bibitem</code> : Removed unnecessary	<code>\normalfont</code> 466
braces 472	<code>\thempfootnote</code> : Added
<code>\nocite</code> : Use <code>\@firstofone</code> 473	<code>\itshape</code> 466
1995-05-08 ltdefs.dtx v1.2k	1995-05-19 ltpictur.dtx v1.1a
<code>\typein</code> : Use <code>\@firstofone</code> 37	General: Support autoloading
1995-05-08 ltdefs.dtx v1.2l	feature 411
<code>\typein</code> : Remove unnecessary	1995-05-20 ltcounts.dtx v1.1b
braces 37	<code>\@definecounter</code> : Streamlined
Replace <code>\def</code> by <code>\let</code> 37	code 148
1995-05-08 ltfsstrc.dtx v2.3n	<code>\@fnsymbol</code> : Allowing both text
<code>\ifnot@nil</code> : Use <code>\@firstofone</code> . . 207	and math 151
1995-05-11 fontdef.dtx v2.2j	<code>\fnsymbol</code> : Streamlined code . . . 150
General: Updates to some plain	1995-05-20 ltcounts.dtx v1.1c
macros 264	<code>\@definecounter</code> : And do it right 148
1995-05-12 ltclass.dtx v1.0n	1995-05-20 ltfloat.dtx v1.1k
<code>\DeclareOption*</code> : Use <code>\toks@</code> to	<code>\@makefnmark</code> : Moved
remove need to double hash	<code>\normalfont</code> back and use
/1557 559	<code>\@textsuperscript</code> 466
1995-05-12 ltfloat.dtx v1.1h	Moved <code>\normalfont</code> to
<code>\@footnotemark</code> : Add <code>\nobreak</code> to	<code>\textsuperscript</code> 466
allow hyphenation. latex/1605 468	<code>\textsuperscript</code> : Use
1995-05-12 ltpictur.dtx v1.0h	<code>\normalfont</code> 467
<code>\pictur@</code> : Macro added for	1995-05-21 ltfssdcl.dtx v2.1t
latex/1355 412	<code>\DeclareMathRadical</code> : Allow for
1995-05-12 ltvers.dtx v1.0e	undefined cs names 243
General: Add autoload docstrip	1995-05-21 ltlists.dtx v1.0f
guards 33	General: Moved to doc.sty
Check for format older than 1	standard 361
year 33	1995-05-21 ltmath.dtx v1.0r
1995-05-13 ltfsstrc.dtx v2.3o	<code>\@sqrt</code> : Use <code>\sqrtsign</code> 356
General: Use single hash mark in	General: Remove <code>\mathhexbox</code>
<code>\DeclareOption</code> 196	from this file 353
1995-05-16 ltfloat.dtx v1.1i	Update some plain macros . . . 348
<code>\@makefnmark</code> : Now use	<code>\lefteqn</code> : Use <code>\rlap</code> 357
<code>\textsuperscript</code> 466	<code>\r@@t</code> : Use <code>\sqrtsign</code> instead of
<code>\textsuperscript</code> : Command	<code>\sqrt</code> 350
added./pr1503 467	1995-05-21 ltoutenc.dtx v1.7h
<code>\thefootnote</code> : Streamlined parts	<code>\@inmathwarn</code> : Added several
of code. 466	<code>\@onlypreamble</code> 106
1995-05-17 ltboxes.dtx v1.0u	1995-05-21 ltoutenc.dtx v1.7j
<code>\@irsbox</code> : Removed surplus	General: Updated some plain
braces 387	macros 118

1995-05-21 ltplain.dtx v1.1j		1995-05-25 ltclass.dtx v1.0p	
General: Moved some code to		<code>\endfilecontents</code> : Delete	
other files	15	<code>\filecontents</code> after preamble	568
1995-05-22 ltplain.dtx v1.1k		1995-05-25 ltfiles.dtx v1.0s	
General: Definitions of <code>\footins</code>		<code>\document</code> : Added check for	
and <code>\footnoterule</code> moved to		<code>\topskip</code> zero	91
<code>ltfloat</code>	30	1995-05-25 ltfiles.dtx v1.0t	
1995-05-22 lttab.dtx v1.1a		<code>\iffileonpath</code> : (CAR) added	
General: Support autoloading		<code>\long</code>	97
feature	389	<code>\document</code> : Corrected typo	91
1995-05-23 ltfssini.dtx v2.2e		<code>\IfFileExists@</code> : (CAR) added	
<code>\newfont</code> : Font assignment made		<code>\long</code>	97
local again.	260	<code>\InputIfFileExists</code> : (CAR)	
1995-05-24 ltdefs.dtx v1.1l		added <code>\long</code>	98
<code>\newif</code> : (DPC) New		<code>\nofiles</code> : (CAR) added <code>\long</code> . .	93
implementation	42	<code>\protected@write</code> : (CAR) added	
1995-05-24 ltdefs.dtx v1.2m		<code>\long</code>	94
<code>\typein</code> : (DPC) New		1995-05-25 ltfloat.dtx v1.1m	
implementation	37	<code>\@savemarbox</code> : (CAR) Resettings	
1995-05-24 ltfloat.dtx v1.1l		moved to hook	462
<code>\@textsuperscript</code> : Command		<code>\@xfloat</code> : (CAR) Resettings	
added.	467	moved to hook	455
General: Moved definition of		1995-05-25 ltlists.dtx v1.0i	
<code>\footins</code> and <code>\footnoterule</code>		<code>\endtrivlist</code> : Macros moved from	
from <code>ltplain</code>	466	<code>ltspace.dtx</code>	370
<code>\textsuperscript</code> : Use		1995-05-25 ltmath.dtx v1.3c classes	
<code>\@textsuperscript</code>	467	<code>\@eqnnum</code> : replace	
1995-05-24 ltfssbas.dtx v3.0a		<code>\reset@font\rmfamily</code> with	
General: (DPC) Make file from		<code>\normalfont</code> (PR 1578)	358
previous file, <code>fam.dtx</code>		1995-05-25 ltspace.dtx v1.2f	
1995/05/20 v2.2d	154	<code>\@vbsphack</code> : (CAR) not used so	
<code>\mathgroup</code> : (DPC) No need to		‘removed’.	80
redefine <code>\newfam</code> as not outer	154	<code>\@vspacer</code> : (CAR) <code>\@restorepar</code>	
1995-05-24 ltfsscnp.dtx v3.0a		added to avoid possible infinite	
General: (DPC) Make file from		tail recursion caused by a typo	
previous file, <code>fam.dtx</code>		in the argument.	83
1995/05/20 v2.2d	218	(CAR) macros modified to be	
1995-05-24 ltfssdcl.dtx v3.0a		more efficient	83
General: (DPC) Make file from		General: Macros moved to	
previous file, <code>latint.dtx</code>		<code>ltlists.dtx</code>	71
1995/05/21 v2.1t	222	1995-05-26 ltdefs.dtx v1.2n	
1995-05-24 ltfssini.dtx v3.0a		<code>\@gobblefour</code> : (CAR) Added	
General: (DPC) Make file from		<code>\longs</code>	44
previous file, <code>lfonts.dtx</code>		1995-05-26 ltmath.dtx v1.0s	
1995/05/23 v2.2e	247	<code>\@eqnnum</code> : Removed <code>\rmfamily</code>	
<code>\cal</code> : (DPC) Remove definition .	263	(PR 1578), replaced	
<code>\mit</code> : (DPC) Remove definition .	263	<code>\reset@font</code> with	
1995-05-24 ltfsstrc.dtx v3.0a		<code>\normalfont</code>	355
General: (DPC) Make file from		1995-05-26 ltpage.dtx v1.0g	
previous file, <code>tracefnt</code>		<code>\ps@plain</code> : removed <code>\rmfamily</code>	
1995/05/16 v2.3o	195	(PR 1578)	476
1995-05-24 ltfsstrc.dtx v3.0b		1995-05-27 ltfssbas.dtx v3.0b	
General: (DPC) Fix		<code>\mathgroup</code> : (FMi) But a need to	
<code>\ProvidesFile</code> usage	195	define <code>\new@mathgroup</code>	154

1995-06-05 fontdef.dtx v2.2k	1995-06-19 ltbibl.dtx v1.1h
General: Moved math commands	\bibtex : Call \@newlabel so
from ltoutenc.dtx. 282	repeated keys produce better
1995-06-05 ltfinal.dtx v1.0r	warning. 472
General: Added \MakeUppercase	1995-06-19 ltclass.dtx v1.0q
and \MakeLowercase 612	\documentclass : Dont redefine
1995-06-05 ltoutenc.dtx v1.7k	\usepackage in compat mode
\@inmathwarn : Removed	for /1634 562
\protected@cmd and replaced	1995-06-19 ltxref.dtx v1.1e
with explicit \noexpand 106	\newlabel : Use \@newlabel to
General: Allowed	share code with \bibtex 331
\ProvideTextCommandDefault	1995-06-28 ltffssini.dtx v3.0b
after the preamble. 108	General: (DPC) Fix
Commented out \textless and	documentation typos 247
\textgreater 115	1995-06-28 ltmath.dtx v1.0t
Moved math commands to	General: minor doc edits 348
fontdef.dtx. 118	1995-07-02 ltplain.dtx v1.1n
Save some tokens in	General: Removed surplus ‘by’ and
\textvisiblespace and	‘=’ in various places 15
\textunderscore 115	\offinterlineskip : Replaced
1995-06-06 ltfinal.dtx v1.0s	1000 by \@m 27
General: Made \MakeUppercase	\showoutput : Use \showoverfull
and \MakeLowercase brace	to save space 30
their argument. 612	\tracingall : Use \showoutput to
1995-06-09 ltoutenc.dtx v1.7l	save space 30
\DeclareTextComposite : Rewrote	1995-07-03 ltdefs.dtx v1.2o
\DeclareTextComposite to	\set@typeset@protect : Use
define the composite as a	\@typeset@protect for init 46
no-argument command rather	1995-07-03 ltfntcmd.dtx v3.3s
than a two-argument	\t@st@ic : Use clean interface for
command. 109	jump 294
1995-06-11 ltspace.dtx v1.2g	1995-07-05 ltfntcmd.dtx v3.3s
\restorecr : (CAR) \relax added	\t@st@ic : Renamed from
to stop silent eating of *. 86	\test@next 293
1995-06-13 ltfinal.dtx v1.0t	1995-07-05 ltspace.dtx v1.2h
General: Add patch level string	\@gnewline : Use \break 76
more carefully 624	\@no@pgbk : Macro replaces \@pgbk
Call \errorstopmode 626	and \@nopgbk 75
1995-06-13 ltpictur.dtx v1.1b	\nopagebreak : Reimplemented
General: Use \ProvidesFile in	both using \@no@pgbk 74
autoload 411	1995-07-09 ltcntrl.dtx v1.0f
1995-06-14 lttab.dtx v1.1b	\@iforloop : Reimplemented using
General: Use \ProvidesFile in	Kabelschacht method 59
autoload 389	\@iwhiledim : Reimplemented
1995-06-15 ltffsbas.dtx v3.0c	using Kabelschacht method 57
General: (DPC) minor	\@iwhilenum : Reimplemented
documentation changes 154	using Kabelschacht method 57
1995-06-15 ltffscmp.dtx v3.0b	\@iwhilesw : Reimplemented using
General: (DPC) minor	Kabelschacht method 57
documentation edits 218	\@tfor : Reimplemented using
1995-06-15 ltffsdcl.dtx v3.0b	Kabelschacht method 59
General: (DPC) minor	1995-07-09 ltlists.dtx v1.0j
documentation changes 222	enumerate : Use \expandafter 375
	itemize : Use \expandafter 375

1995-07-12 ltpictur.dtx v1.1d		1995-08-16 ltcentrl.dtx v1.0g	
General: allow 2e commands in 209 mode. latex/1737	411	\@break@tfor: Made long	59
1995-07-13 ltdefns.dtx v1.0p		\@forloop: Made defs long	58
General: Updates to documentation	36	\@fornoop: Made defs long	58
1995-07-13 ltfiles.dtx v1.0u		\@iforloop: Made defs long	59
General: Updates to docu	88	\@iwhiledim: Made defs long	57
1995-07-13 ltfssbas.dtx v3.0d		Removed \@whilenoop	57
\@defaultsubs: macro added	172	\@iwhilenum: Made defs long	57
\@defaultsubs: macro added	172	Removed \@whilenoop	57
General: minor documentation changes	154	\@iwhilesw: Removed	
\wrong@fontshape: Change a macro not a switch to flag default font substitutions	171	\@whileswnoop	57
1995-07-13 ltmiscen.dtx v1.0z		\@tfor: Made defs long	59
\@centercr: Use \@nobreak	341	1995-08-16 ltfiles.dtx v1.0v	
\@writefile: Added missing percent and use \@relax in the THEN case	337	\document: set \@maxdepth	91
\@xobeyssp: Use \@nobreak	343	set \@do globally	92
General: Improve Documentation	333	set \@topskip globally	91
\enddocument: Set \@setckpt to \@gobbletwo instead of defining it by hand	334	1995-08-24 ltfssbas.dtx v3.0f	
Shorten redefinition of \@bibtex and \@newlabel	335	General: Added autoload code	154
Use \@defaultsubs instead of switch	335	1995-08-24 ltfsstrc.dtx v3.0c	
1995-07-14 ltbibl.dtx v1.1i		General: Macro	
\@bibtex: Remove		\@gobble@font@spec removed	207
\@onlypreamble so still defined in new \@enddocument	472	\@tryis@simple:	214
1995-07-14 ltxref.dtx v1.1g		1995-08-25 ltoutput.dtx v1.1p	
\@newlabel: Remove		General: Support autoloading feature (FMi).	478
\@onlypreamble so still defined in new \@enddocument	331	1995-09-01 lterror.dtx v1.2i	
1995-07-19 ltfssini.dtx v3.0d		General: Add autoload support	60
General: (DPC) TeX2 support	263	1995-09-01 ltplain.dtx v1.1m	
1995-07-20 ltboxes.dtx v1.0v		\@empty: Use \@let to save space	27
\@isavebox: Use \@sbox	380	\@I: Use \@let to save space	26
\@isavepicbox: Use \@sbox	380	1995-09-14 ltplain.dtx v1.1o	
1995-07-21 ltoutput.dtx v1.1o		General: Moved \@multispan to lttab.dtx	15
\@writesetup: Command added	505	1995-09-14 lttab.dtx v1.1c	
New, experimental, versions: need in-lining	505	\@cline: (DPC) New implementation	409
1995-08-09 ltmath.dtx v1.0u		1995-09-15 ltfssini.dtx v3.0e	
General: Added code for class options leqno and fleqn	357	General: (DPC) Modify TeX2 message	263
1995-08-11 ltlength.dtx v1.1b		1995-09-19 ltmiscen.dtx v1.1a	
General: Doc typos fixed for latex/753	153	\@verb: Put \@noligs after \@verbatim@font where it belongs.	346
		1995-10-01 ltfiles.dtx LaTeX2e	
		\@addtofilelist: Macro added	100
		1995-10-02 ltdefns.dtx v1.2q	
		\@ifnch: Use \@let@token for internal/924, save \@reserved@e	50
		\@ifnextchar: Use \@let@token	50
		\@protected@testopt: Macro added	40
		\@testopt: Macro added	39
		\@xargdef: New implementation, using \@test@opt	39

1995-10-03 fontdef.dtx v2.2l		1995-10-16 ltssini.dtx v3.0f	
General: <code>\@@sqrt</code> from patch file		<code>\reset@font</code> : Added <code>\relax</code> after	
for /1701	264	<code>\usefont</code> , as the latter eats up	
1995-10-03 ltdefs.dtx v1.2r		spaces.	261
<code>\typein</code> : Add missing <code>\@typein</code>		1995-10-16 ltmath.dtx v1.0y	
for /1710 (from patch file) . . .	38	<code>\@yeqncr</code> : (DPC) Use <code>\@testopt</code>	
1995-10-03 ltpictur.dtx v1.1e		/1911	356
General: New autoloading code	411	<code>\sqrt</code> : (DPC) Make robust /1808	356
1995-10-04 ltssbas.dtx v3.0g		1995-10-16 ltspc.dtx v1.2j	
General: Modify autoloading code . .	154	<code>\nolinebreak</code> : (DPC) Use	
1995-10-04 ltssstrc.dtx v3.0d		<code>\@testopt</code> /1911	74
General: (DPC) Modify autoloading		<code>\nopagebreak</code> : (DPC) Use	
code	195	<code>\@testopt</code> /1911	74
1995-10-04 lttab.dtx v1.1d		1995-10-16 lthm.dtx v1.0g	
General: Modify autoloading support	389	General: Revert to previous	
1995-10-06 ltfiles.dtx v1.0w		<code>\newtheorem</code> behaviour	434
<code>\@missingfileerror</code> : Autoloading		1995-10-17 ltclass.dtx v1.0r	
error	99	<code>\@providesfile</code> : Delay definition	
1995-10-09 lterror.dtx v1.2j		of <code>\ProvidesFile</code> till <code>ltfinal</code>	559
General: Modify autoloading support	60	<code>\ProcessOptions*</code> : Reset	
1995-10-09 ltoutenc.dtx v1.7m		<code>\CurrentOption</code> for	
<code>\@inmathwarn</code> : Autoloading error . .	107	<code>graphics/1873</code>	561
1995-10-10 ltssbas.dtx v3.0h		1995-10-17 ltdirchk.dtx v1.0l	
<code>\showhyphens</code> : Use <code>\normalfont</code>		General: Modify <code>initex</code> version of	
and make colour safe, and		<code>\ProvidesFile</code>	4
autoloading	175	1995-10-17 ltfinal.dtx v1.0v	
1995-10-10 ltssdcl.dtx v3.0c		<code>\@providesfile</code> : reset macro . . .	626
<code>\non@alpherr</code> : (DPC) autoloading		<code>\reserved@b</code> : reset here after the	
error message	226	<code>\input</code> above	625
1995-10-10 ltplain.dtx v1.1r		1995-10-17 ltplain.dtx v1.1s	
General: Autoloading tracing code . .	15	<code>\eject</code> : Move <code>\supereject</code> to	
1995-10-10 lthm.dtx v1.0f		<code>compat</code> file	28
General: Make <code>\newtheorem</code> ‘only		1995-10-17 lttab.dtx v1.1e	
preamble’	434	<code>\@cline</code> : (DPC) Use <code>\@multicnt</code>	410
1995-10-11 ltoutput.dtx v1.1r		<code>\@multispan</code> : (DPC) Macro	
<code>\clearpage</code> : Added a check so that		added.	410
it does not lose the argument		1995-10-19 ltfinal.dtx v1.0w	
of <code>\twocolumn[...]</code>	491	<code>\@filelist</code> : Move after	
1995-10-16 ltbibl.dtx v1.1j		<code>\reserved@a</code> setting;-)	626
<code>\cite</code> : (DPC) Make robust	472	1995-10-20 ltbibl.dtx v1.1k	
1995-10-16 ltboxes.dtx v1.0w		<code>\@citex</code> : Removed redefined	
General: Clarify <code>makebox</code>		flag	472
description	377	<code>\nocite</code> : Removed redefined	
1995-10-16 ltdefs.dtx v1.2u		flag	473
<code>\@ifstar</code> : (DPC) New		1995-10-20 ltclass.dtx v1.0s	
implementation, for /1910 . . .	50	<code>\@begindocumenthook</code> : Make	
<code>\new@command</code> : (DPC) Use		setting conditional, for	
<code>\@testopt</code> /1911	39	autoloading version	567
<code>\new@environment</code> : (DPC) Use		1995-10-20 ltssbas.dtx v3.0i	
<code>\@testopt</code> /1911	41	General: (DPC) Modify autoloading	
<code>\typein</code> : (DPC) Use <code>\@testopt</code>		code, change <code>\undefined</code> . . .	154
/1911	37	1995-10-20 ltssstrc.dtx v3.0e	
		General: (DPC) Modify autoloading	
		code	195

1995-10-22 ltffsbas.dtx v3.0j	General: (RmS) New size function macro <code>\genb@sfcnt</code> needs to be disabled at <code>\document.</code> . . . 154	<code>\end@dblfloat</code> : (CAR) unify code for double and single versions 458
1995-10-22 ltfsstrc.dtx v3.0f	General: Added ‘genb’ and ‘sgenb’ size functions to support new DC font naming scheme. . . . 195	<code>\end@float</code> : (CAR) unify code for double and single versions . . . 457
1995-10-23 lttab.dtx v1.1f	<code>\settab</code> : (CAR)Ensure that <code>\@hightab</code> increases by at most one 396	1995-10-25 ltidxglo.dtx v1.1d
	<code>\startline</code> : (CAR)Ensure that <code>\@nxttabmar</code> is never larger than <code>\@hightab</code> 394	General: Doc cleanup 469
	<code>\poptabs</code> : (CAR)Ensure that <code>\@curtab</code> is never larger than <code>\@hightab</code> 397	1995-10-25 ltsect.dtx v1.0q
	<code>\tabbing</code> : (CAR)Make <code>\@hightab</code> consistently a local variable . 395	<code>\ subparagraphmark</code> : Use <code>\let</code> not <code>\def</code> to save space. 445
1995-10-24 ltfiles.dtx v1.1a	<code>\document</code> : Removed	1995-10-27 ltpictur.dtx v1.1f
	multiplelabels switch 91	General: Move initialisation to kernel from autoload file . . . 430
	Removed redefined switch . 91	1995-10-31 ltboxes.dtx v1.0x
1995-10-24 ltffsbas.dtx v3.0k	<code>\@defaultsubs</code> : macro removed 172	<code>\@finalstrut</code> : Add <code>\nobreak</code> in horiz mode to allow hyphenation. internal/1931 . 388
	<code>\wrong@fontshape</code> : Make this code inline since it happens only here 171	1995-11-01 fontdef.dtx v2.2m
1995-10-24 ltmiscen.dtx v1.1b	<code>\enddocument</code> : Changed logic for producing warning messages and removed switch 335	General: add <code>\nfss@catcodes</code> for internal/1932 267
	Use <code>\@redefined</code> instead of switch 335	1995-11-01 ltdirchk.dtx v1.0n
1995-10-24 ltxref.dtx v1.1h	<code>\@multiplelabels</code> : Switch for multiplelabels removed 331	General: Initialise <code>\@addtofilelist</code> to <code>\@gobble</code> . 4
	<code>\@newl@bel</code> : Switch for multiplelabels replaced by inline code 331	1995-11-01 ltfinal.dtx v1.0x
	<code>\@redefined</code> : Switch for redefined replaced 330	General: (DPC) Switch meaning of <code>\@addtofilelist</code> for cfg files 617
	<code>\@setref</code> : Switch for redefined renamed 331	1995-11-01 ltffsbas.dtx v3.0m
	<code>\if@multiplelabels</code> : Macro removed 331	<code>\DeclareFontShape@</code> : (DPC) Test for <code>\relax</code> not <code>\undefined</code> , internal/1933 155
1995-10-25 ltalloc.dtx v1.1b	General: General doc improvements 54	1995-11-01 ltffsini.dtx v3.0g
1995-10-25 ltfloat.dtx v1.1n	<code>\endfloatbox</code> : (CAR) macro added: to unify code for double and single versions 459	General: (DPC) Switch meaning of <code>\@addtofilelist</code> for cfg files 263
		1995-11-02 ltffsbas.dtx v3.0n
		<code>\wrong@fontshape</code> : (DPC) Remove extra space with <code>\string</code> for latex/1676 170
		1995-11-02 ltoutenc.dtx v1.7n
		General: Changed internal name <code>\a</code> to <code>\@tabacckludge</code> to protect against redefinition by malicious users. 114
		1995-11-07 ltlists.dtx v1.0k
		<code>\@doendpe</code> : Enclosed <code>\setbox0</code> assignment by a group so that it leaves the contents of box 0 intact. 371
		1995-11-07 ltoutenc.dtx v1.7o
		General: Added <code>\leavevmode</code> at start of <code>\c</code> , otherwise the output routine might be invoked within the macro. . . 118
		Changed <code>\char32</code> to <code>\@xxxii</code> (two tokens less). 119

- Replaced octal number 27 by decimal number 23 to protect against the quote character being active. 119
- Replaced some 0's by \z@ (faster). 119
- 1995-11-10 ltoutput.dtx v1.1s
 \@shipoutsetup: Command removed 505
 \@writsetup: Command removed 505
 In-lined 505
- 1995-11-14 ltclass.dtx v1.0t
 \@unprocessedoptions: Allow empty option 568
 \@loadwithoptions: macro added 562
 \@loadClassWithOptions: macro added 563
 \@RequirePackageWithOptions: macro added 563
- 1995-11-17 ltfssbas.dtx v3.0m
 \@wrong@font@char: (DPC) Macro added. latex/1676 172
 \@define@newfont: Redefine \typeout latex/1676 165
 \@wrong@fontshape: Support \@wrong@font@char latex/1676 170
- 1995-11-17 ltoutenc.dtx v1.7p
 \@UseTextSymbol: Support \@wrong@font@char latex/1676 111
- 1995-11-18 ltoutenc.dtx v1.7q
 \@UseTextSymbol: Modify message slightly 111
- 1995-11-21 fontdef.dtx v2.2n
 General: Incorporate changed figures, as in plain.tex 280
- 1995-11-27 ltfssbas.dtx v3.0n
 \@nfss@catcodes: Reset hash, for definitions in fd files 167
- 1995-11-28 ltfloat.dtx v1.1n
 General: documentation fixes 450
- 1995-11-28 ltfssstrc.dtx v3.0g
 General: documentation fixes 195
- 1995-11-28 ltoutenc.dtx v1.7r
 General: Added math mode checks to text commands. 106
 doc fixes 102
 Renamed \@changed@x@err to \@TextSymbolUnavailable. 106
- 1995-11-29 ltoutenc.dtx v1.7t
 General: Added
 \textasciicircum,
 \textasciitilde,
 \textbackslash, \textbar,
 \textgreater and \textless. 121
 Added \textasciicircum,
 \textasciitilde,
 \textregistered and
 \texttrademark. 115
 Added \textbackslash and
 \textbar. 115, 125
 Added \textless and
 \textgreater. 115, 126
- 1995-12-01 ltoutenc.dtx v1.7u
 General: Made \SS a Default, rather than having the default point to the OT1 definition. 115
- 1995-12-04 ltspc.dtx v1.2k
 \nobreakspace: (Macro added 84
- 1995-12-04 ltspc.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 286
- 1995-12-05 ltdefns.dtx 1.2w
 \@unexpandable@protect: Removed
 \unexpandable@noexpand as never used. internal/1733 45
- 1995-12-05 ltfiles.dtx v1.1c
 \@document: \ignorespaces added for latex/1933 92
- 1995-12-05 ltfloat.dtx v1.1n
 \@textsuperscript: Use \ensuremath for latex/1984. 467
- 1995-12-05 ltoutenc.dtx v1.7v
 \@inmathwarn: Changed
 \@TextSymbolUnavailable text 107
- 1995-12-06 ltfssbas.dtx v3.00
 \@nfss@catcodes: Reset hat, for typeouts etc in fd files 167
- 1995-12-07 ltbibl.dtx v1.1l
 \@citex: Restored name of
 \@G@refundefinedtrue 472
- 1995-12-07 ltfloat.dtx v1.1m
 \@textsuperscript: Move \m@th out of the \ensuremath for latex/1984. 467
- 1995-12-07 ltxref.dtx v1.1i
 \@setref: Switch for redefined restored 331
 \@G@refundefinedtrue: Renamed (back) from \@G@refundefined 330

1995-12-11 ltoutenc.dtx v1.7w	1996-05-18 ltoutenc.dtx v1.7x
General: Modified <code>\copyright</code> . . . 115	General: Produce error if encoding not found. pr/2054 143
1995-12-13 ltdefs.dtx 1.2x	1996-05-21 ltoutenc.dtx v1.7y
<code>\-</code> : Documentation changed. 51	General: Corrected error message (CAR) 143
1996-01-10 ltfiles.dtx v1.1d	1996-05-21 ltsect.dtx v1.0s
<code>\@iffileonpath</code> : Change argument handling to not require doubled hash. latex/2024 97	<code>\@sect</code> : (DPC) Added extra braces for internal/2148 442
1996-01-20 ltidxglo.dtx v1.1e	(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 . . . 442
<code>\makeglossary</code> : Make no-op after use pr/2048 470	<code>\@ssect</code> : (DPC) Added extra braces for internal/2148 445
<code>\makeindex</code> : Make no-op after use pr/2048 470	(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 . . . 444
1996-01-20 ltspace.dtx v1.2m	1996-05-23 ltoutenc.dtx v1.7z
<code>\vspace</code> : Made robust 83	<code>\@strip@args</code> : <code>\expandafter</code> added to match other changes for latex/2133 111
1996-03-25 ltmath.dtx v1.1a	<code>\add@accent</code> : macro added. latex/2133 108
<code>\@ensuredmath</code> : Macro added for amslatex/2104 357	<code>\DeclareTextAccent</code> : Reimplemented using <code>\add@accent</code> to save space latex/2133 108
<code>\ensuremath</code> : Reimplement for amslatex/2104 357	<code>\DeclareTextCompositeCommand</code> : Modified to cope with new <code>\add@accent</code> command: required removal of check for one argument-command . . . 109
1996-04-18 ltpage.dtx v1.0i	1996-05-24 ltoutput.dtx v1.1t
General: Improve documentation 475	<code>\@specialoutput</code> : Check that <code>\@colroom</code> is less than <code>\vsize</code> , indicating that a float has been added 496
1996-04-22 ltmiscen.dtx v1.1c	Cut-off point changed to <code>1.5\baselineskip</code> 496
General: Improve Documentation 333	<code>\@topnewpage</code> : Cut-off point changed to <code>2.5\baselineskip</code> 495
1996-04-22 ltspace.dtx v1.2n	1996-05-25 ltoutput.dtx v1.1u
General: Documentation	<code>\@specialoutput</code> : Correct the above check 496
Improvements 71	1996-06-03 ltmiscen.dtx v1.1d
1996-04-22 lttab.dtx v1.1g	<code>\@verbatim</code> : Exchanged the following two code lines so that <code>\dospecials</code> cannot reset the
<code>\@tabclassz</code> : (DPC) Extra <code>\hskip</code> keeps <code>tabcolsep</code> in empty columns internal/2122 408	
1996-04-23 ltcounts.dtx v1.1d	
General: Documentation	
improvements 146	
1996-04-24 ltfiles.dtx v1.1e	
<code>\document</code> : (DPC) Reset <code>\AtBeginDocument</code> eg for latex/1297 91	
1996-05-08 ltfstrc.dtx v3.0h	
<code>\math@egroup</code> : Use <code>\bgroup</code> instead of <code>\begingroup</code> to match a kernel change made in 1994!! 205	
1996-05-09 ltfntcmd.dtx v3.3t	
<code>\check@icr</code> : Default definitions added 292	
1996-05-17 fontdef.dtx v2.2o	
General: <code>\@@sqrt</code> removed, at last 264, 280	
1996-05-17 ltfiles.dtx v1.1f	
<code>\nofiles</code> : added <code>\write</code> to <code>\protected@write</code> for latex/2146 93	

category code of characters handled by <code>\@noligs</code>	344	<code>\@savemarbox</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code>	462
General: Move setting of verbatim font and <code>\@noligs</code>	333	<code>\@setminipage</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code>	457
<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>	346	<code>\@setnobreak</code> : remove unnecessary <code>\global</code> before <code>\@nobreak...</code>	457
1996-06-10 <code>ltxboxes.dtx</code> v1.0y <code>\@parboxto</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code>	383	1996-07-26 <code>ltxssbas.dtx</code> v3.0p <code>\@DeclareMathSizes</code> : use faster <code>\if</code> test	160
1996-06-10 <code>ltxsect.dtx</code> v1.0t <code>\@sect</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code>	442	<code>\nfss@catcodes</code> : omit <code>\relax</code> as not needed	167
<code>\@sect</code> : (DPC) Changed <code>\endgraf</code> to <code>\@@par</code>	444	1996-07-26 <code>ltxssdcl.dtx</code> v3.0e <code>\init@restore@version</code> : Removed <code>\ifrestore@version</code> switch and replaced by <code>\init@restore@version</code>	226
1996-06-13 <code>ltxdirchk.dtx</code> v1.0r General: documentation improvements mainly from internal/2174	1	1996-07-26 <code>ltxsstrc.dtx</code> v3.0i <code>\init@restore@glb@settings</code> : macro added replacing <code>\if@inmath</code> switch	204
1996-06-14 <code>ltxtab.dtx</code> v1.1h <code>\@tabclassz</code> : (DPC) Change both <code>\z@skip</code> to <code>lsp</code> for <code>latex/2160</code>	408	1996-07-26 <code>ltxlists.dtx</code> v1.0l <code>\@item</code> : Remove unnecessary <code>\global</code> before <code>\@minipage...</code>	372
1996-06-22 <code>ltxspace.dtx</code> v1.2o General: Documentation of problems added	71	Remove unnecessary <code>\global</code> before <code>\@nobreak...</code>	373
1996-07-10 <code>ltxfinal.dtx</code> v1.0y <code>\@toks</code> : Free up memory from scratch registers /2213	625	1996-07-26 <code>ltxmath.dtx</code> v1.1b General: Removed <code>\global</code> before <code>\@ignoretrue</code> in various places.	348
1996-07-19 <code>ltxoutenc.dtx</code> v1.8a <code>\@strip@args</code> : Use char 0 not @ as carrier for <code>\lowercase</code> /2197	111	1996-07-26 <code>ltxmiscen.dtx</code> v1.1e <code>\@ignorefalse</code> : put <code>\global</code> into definition	334
1996-07-26 <code>ltxboxes.dtx</code> v1.0z <code>\if@minipage</code> : put <code>\global</code> into definition	385	<code>\begin</code> : remove <code>\global</code> before <code>\@ignore...</code>	338
1996-07-26 <code>ltxclass.dtx</code> v1.0u <code>\@classoptionslist</code> : made only preamble	554	<code>\end</code> : remove <code>\global</code> before <code>\@ignore...</code>	340
<code>\@unusedoptionlist</code> : made only preamble	555	<code>\ignorespacesafterend</code> : user level macro added	334
1996-07-26 <code>ltxdefns.dtx</code> v1.2y <code>\@reargdef</code> : third arg picked up by <code>\@yargdef</code>	41	1996-07-26 <code>ltxoutput.dtx</code> v1.1v <code>\@testfp</code> : remove <code>\global</code> before <code>\@test...</code>	538
<code>\@renew@command</code> : use <code>\noexpand</code> instead of <code>\string</code>	41	<code>\@xtryfc</code> : remove <code>\global</code> before <code>\@test...</code>	513
use <code>\relax</code> in place of empty arg	41	<code>\@ztryfc</code> : remove <code>\global</code> before <code>\@test...</code>	514
<code>\@renew@environment</code> : use <code>\relax</code> in place of empty arg	42	General: put <code>\global</code> into definition	488
1996-07-26 <code>ltxfloat.dtx</code> v1.1n <code>\@endfloatbox</code> : remove unnecessary <code>\global</code> before <code>\@minipage...</code>	459	remove <code>\global</code> before <code>\@test...</code>	488, 489
		<code>\clearpage</code> : add number of missing percents	491

1996-07-26 ltplain.dtx v1.1t		1996-09-29 ltsect.dtx 1.0w	
\sh@ft: replace \dimen\z@ by		\@noskipsectrue: Added	
\dimen@	29	documentation	439
1996-07-26 ltsect.dtx v1.0u		1996-09-30 ltoutput.dtx v1.1y	
\@starttoc: removed \global		\newpage: Checks for noskipsec	
before \@nobreak	446	and inlabel removed pending	
\@xsect: Removed \global before		further tests	492
\@nobreak	443	1996-10-04 ltclass.dtx v1.0v	
1996-07-26 ltspace.dtx v1.2p		\RequirePackageWithOptions:	
\if@nobreak: put \global inside		Reset \@unprocessedoptions	
definition	76	for /2269	563
1996-07-27 ltffsbas.dtx v3.0q		1996-10-05 ltfiles.dtx v1.1h	
General: \if@inmath switch		\@clubpenalty: Added setting its	
removed	165	value	90
1996-07-27 ltspace.dtx v1.2q		1996-10-08 ltfntcmd.dtx v3.3u	
General: Further documentation of		\DeclareTextFontCommand:	
problems	71	Removed \check@icr when in	
1996-07-27 ltspace.dtx v1.2r		vmode since it causes various	
General: Correct documentation of		errors (see pr/2157)	290
problems	71	1996-10-21 lttab.dtx v1.1i	
1996-08-02 ltfloat.dtx v1.1o		\@array: Use	
\@xympar: Remove \global before		\set@typeset@protect	401
\@ignore	463	General: Moved the code	
1996-08-02 ltsect.dtx v1.0v		associated with \@mkpream into	
\@afterheading: Removed		the group provided by the box,	
\global before \@nobreak	445	for robustness (latex/2183)	401
1996-08-02 ltspace.dtx v1.2s		\multicolumn: Make	
\@Esphack: Remove \global		\multicolumn long	
before \@ignore	79	(latex/2180)	403
1996-08-25 ltffsbas.dtx v3.0r		\tabbing: Moved the \indent so	
\@nfss@catcodes: Reset the acute,		that the \everypar can remove	
grave and double quote chars		it when necessary; this is	
as well	167	needed because the code for	
1996-09-21 ltoutput.dtx v1.1w		items in lists has changed (see	
\@writesetup: Added		pr/22111)	395
\@parboxrestore and made		1996-10-23 ltlists.dtx v1.0m	
consequent deletions: wait for		\@item: \@nobreak moved into	
the howls of protest	505	the \everypar and not	
1996-09-25 ltdirchk.dtx v1.0t		executed unconditionally, see	
General: Move ltxcheck to separate		above	373
file	14	\kern changed to	
1996-09-28 ltmiscen.dtx v1.1f		\setbox	372
\@xobeysp: Moved to ltspace.dtx	343	Added setting of \@clubpenalty	
1996-09-28 ltspace.dtx v1.2t		and set \@nobreakfalse only	
\@xobeysp: Moved from		when necessary	373
ltmiscen.dtx and redefined to		1996-10-23 ltsect.dtx v1.0x	
use \@nbreakspace	84	\@xsect: Replaced \hskip	
1996-09-29 ltfiles.dtx v1.1g		with \setbox as used in	
\document: Added disabling of		\@afterheading	443
\@nodocument	92	1996-10-24 ltboxes.dtx v1.1a	
1996-09-29 ltoutput.dtx v1.1x		\@arrayparboxrestore: Added	
\newpage: Checks for noskipsec		local settings of flags:	
and inlabel added	492	dangerous!	384

<code>\@iiiminipage</code> : Use it or lose it (<code>\@setminpage</code>): Frank will want to lose it	385	1996-10-31 <code>ltlists.dtx v1.0p</code>	<code>\@trivlist</code> : Added check for missing item in outer list . . .	369
1996-10-24 <code>lfloat.dtx v1.1p</code>		1996-10-31 <code>ltsect.dtx v1.0y</code>	General: Corrected and tidied documentation; removed long lines	438
<code>\@floatboxreset</code> : Added local settings of flags: dangerous! .	457	1996-11-03 <code>ltplain.dtx v1.1w</code>	<code>\dotfill</code> : Saved tokens by using <code>\hb@xt@</code>	30
<code>\@marginparreset</code> : Added local settings of flags: dangerous! .	463	1996-11-04 <code>lterror.dtx v1.2m</code>	<code>\@nodocument</code> : Always define <code>\@nodocument</code> in kernel, so that it can be cleared by <code>\document</code> .	66
<code>\@xfloat</code> : Added <code>\@nodocument</code> to trap floats in the preamble .	454	1996-11-04 <code>ltlists.dtx v1.0q</code>	<code>\@trivlist</code> : Moved check for missing item: only checked when not inlabel flag is false	369
1996-10-24 <code>ltoutput.dtx v1.1z</code>		1996-11-05 <code>ltfiles.dtx v1.1i</code>	<code>\nofiles</code> : Standard <code>\if@nobreak</code> test added	93
<code>\@addtocurcol</code> : Added <code>\nobreak</code> , etc as appropriate	518, 522	1996-11-09 <code>ltmath.dtx v1.1c</code>	<code>\@ensuredmath</code> : Made long, as it was before. /2104	357
<code>\@specialoutput</code> : Added <code>\nobreak</code> as appropriate . . .	498	1996-11-18 <code>ltfssbas.dtx v3.0s</code>	<code>\define@newfont</code> : (DPC) lowercase fd file names. internal/1044	166
<code>\@topnewpage</code> : Added <code>\@nodocument</code> to trap <code>\twocolumn</code> in the preamble	493	1996-11-18 <code>ltoutenc.dtx v1.8d</code>	General: (DPC) lowercase external file names. internal/1044 . . .	143
<code>\newpage</code> : Better checks for noskipsec and inlabel added, plus nobreak	492	1996-11-20 <code>fontdef.dtx v2.2p</code>	General: lowercase fd and enc.def file names /1044	264
1996-10-25 <code>ltlists.dtx v1.0n</code>		1996-11-20 <code>ltvers.dtx v1.0f</code>	General: Check for old format modified /2319	33
<code>\endtrivlist</code> : Change <code>\indent</code> to <code>\leavevmode</code>	370	1996-11-23 <code>ltoutenc.dtx v1.8e</code>	General: Corrected description . .	103
Reset flags explicitly	370	1996-11-28 <code>ltvers.dtx v1.0g</code>	General: Check for old format modified /2319	33
1996-10-25 <code>ltoutput.dtx v1.2a</code>		1996-12-06 <code>ltdirchk.dtx v1.0u</code>	<code>\IfFileExists</code> : *** removed from various messages for GNU Make. internal/2338	10
<code>\newpage</code> : Reset all flags explicitly	492	1996-12-06 <code>lfloat.dtx v1.1r</code>	<code>\caption</code> : Call <code>\@setminpage</code> if needed. latex/2318	453
1996-10-26 <code>ltlists.dtx v1.0o</code>		1996-12-06 <code>ltfssini.dtx v3.0h</code>	General: (DPC) Remove *** from messages internal/2338	263
<code>\endtrivlist</code> : Correct typo	370			
1996-10-27 <code>ltoutenc.dtx v1.8c</code>				
<code>\@strip@args</code> : Removed macro .	109			
General: Added <code>\r A</code>	119			
Added <code>\textasteriskcentered</code>	115, 125			
Corrected syntax descriptions .	103			
Removed <code>\aa</code> and <code>\AA</code>	115, 119, 121			
1996-10-28 <code>ltplain.dtx v1.1u</code>				
General: (CAR) More doc changes	15			
<code>\dotfill</code> : Removed math mode .	30			
1996-10-29 <code>ltplain.dtx v1.1v</code>				
<code>\dotfill</code> : Got arithmetic correct (CAR)	30			
1996-10-29 <code>ltspace.dtx v1.2u</code>				
<code>\@gnewline</code> : Added macro	76			
<code>\@no@lnbk</code> : Macro replaces <code>\@lnbk</code> and <code>\@no@lnbk</code>	75			
<code>\@</code> : Corrected and rationalised code	75			
<code>\@nolinebreak</code> : Reimplemented both using <code>\@no@lnbk</code>	74			
1996-10-31 <code>ltfinal.dtx v1.0z</code>				
General: Added extra <code>\lcode</code> , hoping it does no harm in T1 (pr/1969)	616, 623			

1996-12-17 ltclass.dtx v1.0w	Modified <code>\underscore</code> , removing <code>\mathunderscore</code> .	116
<code>\g@addto@macro</code> : Use <code>\begingroup</code> to save making a mathord . . .		567
1996-12-20 ltsect.dtx v1.0z	General: Added <code>\leavevmode</code> to <code>\textunderscore</code>	116
<code>\dottedtocline</code> : Added <code>\nobreak</code> for latex/2343 . . .		448
1997-01-08 fontdef.dtx v2.2q	General: Added ‘hex index tabs’ .	122
General: Use <code>\DeclareMathDelimiter</code> to set delimiter codes		273
<code>\mathparagraph</code> : Define using <code>\DeclareMathSymbol</code>		282
1997-01-08 ltfiles.dtx v1.1j	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
<code>\include</code> : reset <code>\deadcycles</code> latex/2365		95
1997-01-08 ltmath.dtx v1.1d	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\root</code> : (DPC) Remove spurious space tokens from plain \TeX definition /2359		350
1997-02-05 ltclass.dtx v1.0x	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
<code>\g@addto@macro</code> : missing percent /2402		567
1997-02-21 ltlists.dtx v1.0r	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\item</code> : <code>\ifvoid</code> check added for <code>\noindent</code> . latex/2414		372
1997-03-21 ltcounts.dtx v1.1e	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\fnsymbol</code> : Use <code>\mathsection</code> and <code>\mathparagraph</code> . latex/2445		150
1997-04-14 ltfiles.dtx v1.1k	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\document</code> : Set the document space factor defaults. latex/2404		91
<code>\normalsfcodes</code> : Macro added (from patch file) latex/2404 . . .		93
1997-04-14 ltoutput.dtx v1.2b	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\writsetup</code> : Call <code>\normalsfcodes</code> (from patch file) latex/2404		507
Move <code>\label</code> and <code>\index</code> (from patch file)		507
1997-04-24 ltbibl.dtx v1.1m	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\citex</code> : <code>\@empty</code> to avoid primitive error on empty cite keys. latex/2432		472
1997-04-30 ltoutenc.dtx v1.9a	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
General: Changed <code>\textsc</code> to <code>\scshape</code>		116
Introduced <code>\textcopyright</code> and modified <code>\copyright</code>		115
Introduced <code>\textcopyright</code> and modify <code>\copyright</code>		116
Modified <code>\textunderscore</code> , removing <code>\mathunderscore</code> . . .		116
1997-04-30 ltoutenc.dtx v1.9b	General: Added <code>\leavevmode</code> to <code>\textunderscore</code>	116
1997-05-04 ltoutenc.dtx v1.9c	General: Added ‘hex index tabs’ .	122
General: Added TS1 encoding v2.2.beta		128
1997-05-07 ltoutenc.dtx v1.9d	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
1997-05-07 ltspace.dtx v1.2v	General: Added <code>\newcommand</code> to <code>\textcompwordmark</code>	116
<code>\newline</code> : Made completely robust.		76
1997-05-29 ltfstsrc.dtx v3.0j	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: Replaced <code>\</code> by <code>\MessageBreak</code> , as suggested by Donald Arseneau.		197
1997-05-29 ltlogos.dtx v1.1f	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
\LaTeXe : Added <code>\m@th</code> so that the \LaTeXe 2 ϵ logo works with non-zero values of <code>\mathsurround</code>		87
1997-06-16 ltdirchk.dtx v1.0v	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: documentation improvements mainly from internal/2520		1
1997-06-16 ltfloat.dtx v1.1s	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: documentation fixes . . .		450
1997-06-16 ltfntcmd.dtx v3.3v	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: Fix typo in documentation.		288
1997-08-05 ltoutenc.dtx v1.9e	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: Corrected order of arguments in <code>\UseTextSymbol</code> example.		103
1997-08-29 ltoutenc.dtx v1.9f	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
General: Added OT4 encoding, provided by Marcin Woliński.		102
1997-09-09 ltdefns.dtx v1.2z	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
<code>\providecommand</code> : Use <code>\begingroup</code> to avoid generating math ords if used in math mode. pr/2573		43
1997-09-15 ltpictur.dtx v1.1g	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
<code>\getcirc</code> : Warn if lines become invisible pr/2524		427
<code>\picture@warn</code> : Macro added pr/2524		427
<code>\@sline</code> : Warn if lines become invisible pr/2524		418
1997-10-06 ltcounts.dtx v1.1f	General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	116
<code>\@Roman</code> : Change <code>\@Roman</code> to be fully expandable, so that the		

- result is written properly to files. 150
- `\@slowromancap`: Macro added. . 150
- 1997-10-08 `ltlogos.dtx v1.1h`
`\LaTeX`: Simplify macro (force loading of suitable math fonts once). 87
- 1997-10-10 `ltclass.dtx v1.0y`
`\endfilecontents`: `\@currentenv` in banner 571
`\reserved@c` not `\verbatim@out` to save a csname 570
Check for text before or after `\end` environment. `latex/2636` 571
Use `\@gobbletwo` 570
- 1997-10-17 `lftntcmd.dtx v3.3w`
`\check@nocorr@`: Check for vertical mode moved here, from `\DeclareTextFontCommand` (see `PR/2646`). 292
`\DeclareTextFontCommand`: Reinstalled `\check@icr` as check is now done in `\check@nocorr@` (see `PR/2646`). 290
- 1997-10-20 `ltfinal.dtx v1.1a`
`\@uclclist`: Removed `\aa` and `\AA` from `\@uclclist` as these are macros. 623
- 1997-10-21 `ltdefs.dtx v1.2z1`
`\renew@command`: Use `\begingroup`/`\endgroup` rather than braces for grouping, to avoid generating empty math atom. 41
- 1997-10-21 `ltfssbas.dtx v3.0t`
`\define@newfont`: Move `\makeatletter` to `\nfss@catcodes`. 166
`\nfss@catcodes`: Moved `\makeatletter` from `\try@load@font@shape`. . . . 167
- 1997-11-09 `ltoutput.dtx v1.2c`
`\@specialoutput`: Remove incorrect code: only one `\@emptycol` is needed here . . 496
`\@topnewpage`: Documentation of `vsize` check enhanced 493
- 1997-11-13 `ltfssdcl.dtx v3.0f`
`\DeclareSymbolFont`: (DPC) Really update `\group@list` dont leave new version in `\toks@`. `latex/2661` 230
- `\stepcounter`: (DPC) Remove as never used. (Re)defined in `ltcounts` 224
- 1997-11-19 `ltfloat.dtx v1.1t`
`\@footnotetext`: Missing percent, again 467
- 1997-11-19 `ltoutput.dtx v1.2d`
`\@vtryfc`: Reindent code, to be understandable(DPC). 512
- 1997-11-20 `ltfssdcl.dtx v3.0g`
`\document@select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a csname per math version as no reset list) 226
`\select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a csname per math version as no reset list) 224
- 1997-11-23 `ltoutenc.dtx v1.9g`
General: Use `\textperthousand`, `\textpertenthousand` and `\textfractionsolidus` not `\textpermill`, `\textpertenmill` and `\textfraction`. `/2673` 128
- 1997-12-17 `ltoutenc.dtx v1.9h`
General: Added `\textperthousand` and `\textpertenthousand` . . 120, 121
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 `\oalign`. 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 . 266
- 1997-12-31 `ltoutenc.dtx v1.9k`
General: Further correction 103

1998-01-12 ltoutenc.dtx v1.9k	1998-04-18 fontdef.dtx v2.2w
General: Added <code>\ProvidesPackage</code>	General: Reinsert symbol def for /
for textcomp.sty 102	char. 274
Adding missing braces and	1998-05-07 ltclass.dtx v1.1b
<code>\ushape</code> 130	<code>\@fileswithoptions</code> : Modify help
1998-01-16 ltoutenc.dtx v1.9m	message for latex/2805 566
General: fixed decimal codes.	1998-05-18 lttab.dtx v1.1j
latex/2734 126	<code>\@endpbox</code> : Use <code>\setlength</code> to set
1998-03-04 ltdefns.dtx v1.2z2	<code>\hspace</code> , so that the changes in
<code>\@xargdef</code> : Unnecessary	the calc package apply here. 410
<code>\expandafter</code> removed:	<code>\tabular*</code> : Use <code>\setlength</code> , so
pr/2758 39	that calc extensions apply. . . 400
1998-03-05 ltoutenc.dtx v1.9n	1998-05-20 ltfinal.dtx v1.1b
General: Added masc/fem ords as	General: Set up lccodes before
in pr/2579 116	loading hyphenation files:
1998-03-20 ltdefns.dtx v1.2z3	pr/2639 616
<code>\@thirdofthree</code> : Macro added . . 44	Set up uc/lccodes after loading
1998-03-20 ltoutenc.dtx v1.9o	hyphenation files: pr/2639 . . 622
General: Documentation added	1998-05-28 lterror.dtx v1.2n
about order of decls 105	<code>\@notdefinable</code> : Added message
Documentation added for	re ‘end..’ pr/1555 65
pr/2783 104	1998-06-04 ltboxes.dtx v1.1c
<code>\UndeclareTextCommand</code> : Macro	<code>\@rule</code> : Support calc-expressions 386
added for pr/2783 113	1998-06-12 ltoutenc.dtx v1.9p
1998-03-20 lttextcomp.dtx v1.9o	General: Corrected 130 and 131,
General: Added various	see pr/2834 131
<code>\UndeclareTextCommand</code>	Renamed <code>\textmacron</code> pr/2840 132
declarations for pr/2783 . . . 324	1998-06-12 ltoutenc.dtx v1.9q
Load decls after defaults for	<code>\add@accent</code> : Explicitly set
speed. 324	<code>\spacefactor</code> after <code>\accent</code>
1998-03-21 ltclass.dtx v1.0z	(pr/2877) 109
General: Added to documentation	1998-06-12 lttextcomp.dtx v1.9p
of filecontents 550	General: Renamed <code>\textmacron</code>
1998-03-21 ltclass.dtx v1.1a	pr/2840 320
<code>\@providesfile</code> : Allow &.	1998-06-18 lttab.dtx v1.1k
Internal/2702 559	General: Small addition to
General: Correct to new	documentation 389
onlypreamble command list . 582	1998-07-06 lttab.dtx v1.1l
1998-03-25 ltfssbas.dtx v3.0u	General: Small correction to
<code>\showhyphens</code> : Suppress	documentation 389
unnecessary error when used in	1998-08-17 ltboxes.dtx v1.1e
preamble 175	General: (RmS) Minor
1998-04-11 fontdef.dtx v2.2t	Documentation fixes. 376
General: Added <code>\mathring</code> accent	1998-08-17 ltclass.dtx v1.1c
(pr/2785) 280	General: (RmS) Minor
1998-04-15 fontdef.dtx v2.2u	documentation fixes. 550
General: Use new syntax for	1998-08-17 ltdirchk.dtx v1.0w
<code>\DeclareMathDelimiter</code> . . . 273	General: (RmS) Documentation
1998-04-15 ltfssdcl.dtx v3.0h	improvements. 1
<code>\@xxDeclareMathDelimiter</code> :	1998-08-17 ltfntcmd.dtx v3.3x
Macro added (pr/2662) 240	General: (RmS) Minor
1998-04-17 fontdef.dtx v2.2v	documentation fixes. 288
General: Reinsert symbol defs for <	
and > chars. 274	

1998-08-17 ltffsbas.dtx v3.0v		<code>\@newenvb</code> : made long and brace	
General: (RmS) Documentation		optional arg. latex/2896	42
fixes.	154	<code>\@testopt</code> : made long and brace	
1998-08-17 ltffsdcl.dtx v3.0i		optional arg. latex/2896	39
General: (RmS) Corrected minor		1999-01-07 ltdefns.dtx v1.3b	
glitches in changes entries.	222	<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.	247	General: Hackery to allow using	
1998-08-17 ltlogos.dtx v1.1i		fontenc several times	145
General: (RmS) Minor		Hackery to temp support cyrillic	
documentation fixes.	87	uc/lc	143
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.	348	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.	333	DPC /2942	40
1998-08-17 ltspace.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 #3	40
General: (RmS) Minor		1999-02-12 tttextcomp.dtx v3.0j	
documentation fixes.	284	<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)	297
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	143
parsing (pr/2884)	401	1999-03-01 ltdefns.dtx v1.3e	
1999-01-05 fontdef.dtx v2.2x		<code>\@ifnextchar</code> : remove extra	
General: Need special protection		<code>\long</code> . internal/2967	50
for character > in <code>\changes</code>		1999-04-15 ltpictur.dtx v1.1h	
entry.	264	<code>\@getlarrow</code> : Replaced octal	
1999-01-06 ltffsbas.dtx v3.0w		number, CAR	419
<code>\DeclareFontEncoding</code> : Added		<code>\@upvector</code> : Replaced octal	
<code>\LastDeclaredEncoding</code> to		number, CAR	420
support cyrillic integration		General: Replaced octal number,	
(pr/2988)	158	CAR	419, 420
<code>\LastDeclaredEncoding</code> : Added		Replaced octal numbers, CAR	411
<code>\LastDeclaredEncoding</code> to		1999-04-19 ltfloat.dtx v1.1u	
support cyrillic integration		<code>\caption</code> : Made caption an error	
(pr/2988)	158	outside a float: latex/2815	453
1999-01-06 ltoutenc.dtx v1.9r		1999-04-27 ltboxes.dtx v1.1f	
<code>\@strip@args</code> : New impl for		<code>\@parboxto</code> : (CAR) Changed	
latex/2930	111	<code>\@empty</code> to <code>\relax</code> as flag for	
General: Minor documentation		natural width: pr/2975	383
fix.	130	1999-04-29 ltdefns.dtx v1.3f	
1999-01-06 ltoutput.dtx v1.2e		<code>\@yargd@f</code> : Full expansion and	
<code>\@makecol</code> : Added negative vskip,		conversion needed for digit in	
as when processing outputbox		new version, see pr/3013	40
below: suggested by Fred		New macro added	40
Bartlett pr/2892	501	1999-06-10 ltoutenc.dtx v1.9u	
1999-01-07 ltdefns.dtx v1.3a		General: Ensure that we also	
<code>\@ifnextchar</code> : made long	50	forget old options (pr/2888)	145

1999-06-12 ltoutenc.dtx v1.9v		<code>\leftmark</code> : Use <code>\@empty</code> instead of brace group (pr/3203).	477
General: Extend <code>\@uclclist</code> only once	144	<code>\markright</code> : Reimplementation to fix expansion error (pr/3203).	476
1999-10-09 ltmath.dtx v1.1e		<code>\rightmark</code> : Use <code>\@empty</code> instead of brace group (pr/3203).	477
<code>\active@math@prime</code> : Macro added, see PR 3104.	353	2000-06-02 ltpage.dtx v1.0k	
<code>\prime@s</code> : Introduce <code>\active@math@prime</code>	353	<code>\@markright</code> : Small adjustment to give slightly less expansion, CAR	477
1999-10-09 ltoutput.dtx 1.2f		<code>\markright</code> : Small adjustment to give slightly less expansion, CAR	476
<code>\@activechar@info</code> : Reset definition of active prime character (used in math mode)	505	Tidied 1.0j reimplementation, CAR	476
1999-10-28 ltoutenc.dtx v1.9w		2000-07-11 ltmiscen.dtx v1.1j	
<code>\add@accent</code> : Give <code>\accent@spacefactor</code> a default definition (pr/3084)	109	<code>\enddocument</code> : Fix typo in warning	335
1999-12-08 ltoutenc.dtx v1.9x		2000-07-12 ltoutput.dtx 1.2g	
General: Changed <code>\CYRRHOOK</code> and <code>\cyrrhook</code> to <code>\CYRRHK</code> and <code>\cyrrhk</code> as name changed in the cyrillic bundle for naming consistency with other “hook” glyphs.	143	General: Ensure that rule is in <code>\normalcolor</code>	544
2000-01-07 ltmiscen.dtx v1.1h		2000-07-12 ltoutput.dtx 1.2i	
<code>\@verbatim</code> : Disable hyphenation even if the font allows it.	344	<code>\@makecol</code> : Removed negative <code>vskip</code> , as it gives unacceptable results when the depth is large: pr/3189	501
2000-01-15 ltpictur.dtx v1.1i		2000-07-19 ltoutput.dtx v1.2h	
<code>\@upvector</code> : Removed space at end-of-line, CAR	420	<code>\@writsetup</code> : Reset and restore <code>\@if@newlist</code> for internal/3231	506
2000-01-30 ltfntcmd.dtx v3.3y		2000-08-23 ltfinal.dtx v1.1c	
<code>\DeclareTextFontCommand</code> : Use <code>\hmode@bgroup</code> now (pr/3160)	290	General: Fix typo in warning	618
2000-01-30 ltoutenc.dtx v1.9y		2000-08-30 ltoutenc.dtx v1.91	
General: Use <code>\hmode@bgroup</code> where applicable (pr/3160)	118–120, 125, 127, 128, 130	<code>\@use@text@encoding</code> : Rearranged but no change to final code, CAR (pr/3160)	111
<code>\add@accent</code> : Use <code>\hmode@bgroup</code> where applicable (pr/3160)	108	<code>\add@accent</code> : Rearranged but no change to final code, CAR (pr/3160)	108
<code>\hmode@bgroup</code> : Macro added	109	2000-09-01 ltfinal.dtx v1.1d	
2000-01-30 ltoutenc.dtx v1.9z		<code>\errhelp</code> : Set error help empty at very end (pr/449 done correctly).	625
<code>\@use@text@encoding</code> : Macro reimplemented (pr/3160)	111, 112	2000-09-24 ltfloating.dtx v1.2b	
<code>\add@accent</code> : Macro reimplemented (pr/3160)	108	<code>\end@dblfloat</code> : FMI: use output routine to defer float	458
<code>\hmode@start@before@group</code> : Macro added (pr/3160)	112	2000-09-24 ltoutput.dtx v1.2b	
2000-05-19 ltmiscen.dtx v1.1i		<code>\@doclearpage</code> : FMI: ensure <code>\doclearpage</code> is called again until all floats are output.	499
<code>\enddocument</code> : Reset <code>\AtEndDocument</code> for latex/3060	334	2000-09-24 ltoutput.dtx v1.2n	
2000-05-26 ltpage.dtx v1.0j		<code>\@addtocurcol</code> : FMI: test for wide float was in wrong place	517
<code>\@markright</code> : Reimplementation to fix expansion error (pr/3203).	477		

2001-01-07 ltoutput.dtx v1.2j		2002-06-16 ltoutenc.dtx v1.95	
\@writsetup: And do it in the		General: Added \textbardbl	
right macro (pr/3286)	506	(pr/3400)	125
2001-02-16 ltxref.dtx v1.1k		Added default for \textbardbl	
\@newl@bel: Added an extra		(pr/3400)	115
grouplevel (PR3250), jlb . . .	331	2002-06-17 ltoutenc.dtx v1.95	
2001-05-25 ltclass.dtx v1.1d		General: Corrected \c for T1	
\@providesfile: Explicitly set		(pr/3442)	120
catcode of \endlinechar to 10		Definition of \textexclamdown	
(pr/3334)	559	changed (pr/3368)	118
2001-05-25 ltdirchk.dtx v1.0x		Definition of	
General: Explicitly set catcode of		\textquestiondown changed	
\endlinechar to 10 (pr/3334) .	4	(pr/3368)	118
2001-05-28 ltoutenc.dtx v1.93		2002-06-18 ltoutenc.dtx v1.95	
General: Added composites for		General: Changed def for	
compatibility with T1,		\textregistered to avoid	
pr/3295	120	small caps (pr/3420)	116
Changed the effect of \.\i,		2002-10-01 lfloat.dtx v1.1v	
pr/3295	122	\thempfootnote: Use braces	
2001-06-02 fontdef.dtx v2.2y		around \itshape to keep font	
General: Provide default cfg files		change local (pr/3460).	466
(pr/3264)	283	2002-10-02 ltfssbas.dtx v3.0x	
2001-06-04 fontdef.dtx v2.2z		\DeclareFontSubstitution:	
General: Guard against math		Adding	
active equal and pipe sign in		\LastDeclaredEncoding	
\models (pr/3333)	279	introduced a bug as on some	
Guard against math active equal		occasions that macro name was	
sign in \Relbar (pr/3333) . . .	279	stored in the internal lists	
2001-06-04 ltclass.dtx v1.1e		instead of the actual encoding.	
\@providesfile: But only if it is a		(pr/3459)	158
char (pr/3334)	559	2002-10-28 ltlists.dtx v1.0s	
2001-06-04 ltdirchk.dtx v1.0y		\endtrivlist: Check for math	
General: But only if it is a char		mode (pr/3437)	370
(pr/3334)	4	2002-10-28 ltoutenc.dtx v1.96	
2001-06-04 ltpictur.dtx v1.1j		General: coding change, to follow	
\@sline: Don't warn for exactly		bug fix by DEK in plain.tex	
zero pr/3318	418	(pr/3469)	119, 128
2001-06-04 ltvers.dtx v1.0i		2002-12-13 ltbibl.dtx v1.1n	
General: Check for old format		\@citex: Added \leavevmode in	
disabled	33	case citation is at start of	
2001-06-05 ltoutenc.dtx v1.94		paragraph (pr/3486)	472
General: Text composite		2003-01-01 ltfntcmd.dtx v3.3z	
Commands need kludges for ‘,		General: Code checked and	
– see tlb1903.lvt	120	documentation extended by	
2001-08-26 ltclass.dtx v1.1f		Chris	290
\@providesfile: Readded setting		2003-05-18 ltbibl.dtx v1.1o	
of space char (pr/3353)	559	\@nocite: Check if we are after	
2002-02-24 ltplain.dtx v1.1x		\document	473
\loggingall: Macro added	30	2003-08-27 ltpictur.dtx v1.1k	
\loggingoutput: Macro added . .	30	\@bezier: added missing	
\showoutput: Use newly added		displacement pr/3566	432
\loggingoutput	30	\@sline: check for \@linechar	
\tracingall: Use newly added		being empty pr/3570	418
\loggingoutput	30		

2003-10-13 ltfinal.dtx v1.1e		2004-02-07 ltoutput.dtx v1.2l	
General: Added extra <code>\lccode</code> for		<code>\@docclearpage</code> : Empty kludgeins	
<code>\-</code> and <code>\textcompwordmark</code> .	617	box if necessary, pr/3528 . . .	500
2003-12-16 ltoutput.dtx v1.2k		2004-02-13 ltoutenc.dtx v1.99e	
<code>\@makecol</code> : Ensure that <code>\@elt</code> has		General: Documentation fixes:	
a defined state (pr/3586) . . .	502	typos	102
2003-12-30 ltpictur.dtx v1.1j		2004-02-15 ltbibl.dtx v1.1q	
<code>\@getcirc</code> : issue warning if circle		<code>\@cite@ofmt</code> : Added hook with	
size can't be met pr/3473 . .	427	default value <code>\hbox</code>	474
2004-01-03 ltoutenc.dtx v1.99b		<code>\@citex</code> : Changed to use a hook	
General: Added		with default value <code>\hbox</code> . . .	473
<code>\textogonekcentered</code>		2004-02-15 ltspace.dtx v1.3a	
(pr/3532)	120	<code>\nbreakdashes</code> : (Added	
Added composites for <code>\k</code>		spacefactor setting	84
(pr/3532)	125	2004-10-20 ltoutput.dtx v1.2m	
Use <code>\ooalign</code> for <code>\k</code> (pr/3532)	120	<code>\@makecol</code> : Removed dead code .	501
2004-01-04 ltbibl.dtx v1.1p		2005-07-27 ltfsdcl.dtx v3.0j	
<code>\nocite</code> : Changed error message	473	<code>\DeclareMathAlphabet</code> : (MH)	
2004-01-04 ltoutenc.dtx v1.99c		Make document commands	
General: More adjustments for		robust	232
ogonek (pr/3532)	120	<code>\DeclareSymbolFontAlphabet</code> :	
2004-01-23 ltdefs.dtx v1.1g		(MH) Make document	
<code>\@newenva</code> : Use kernel version of		commands robust	244
<code>\@ifnextchar</code> (pr/3501)	42	<code>\new@mathalphabet</code> : (MH) Make	
<code>\@testopt</code> : Use kernel version of		document commands robust	233
<code>\@ifnextchar</code> (pr/3501)	40	<code>\non@alpherr</code> : (MH) Change	
<code>\@xargdef</code> : Use kernel version of		because command is now	
<code>\@ifnextchar</code> (pr/3501)	39	properly robust	226
<code>\@xdblarg</code> : Use kernel version of		<code>\SetMathAlphabet</code> : (MH) Make	
<code>\@ifnextchar</code> (pr/3501)	51	document commands robust	234
2004-01-23 ltdefs.dtx v1.3g		2005-09-27 ltoutenc.dtx v1.99g	
<code>\kernel@ifnextchar</code> : Added		General: Replace <code>\sh@ft</code> by	
macro (pr/3501)	50	<code>\ltx@sh@ft</code>	118, 120, 127
2004-01-28 ltclass.dtx v1.1g		2005-09-27 ltplain.dtx v1.1y	
<code>\@providesfile</code> : Use kernel		<code>\ltx@sh@ft</code> : New macro	29
version of <code>\@ifnextchar</code>		<code>\sh@ft</code> : Macro no longer used but	
(pr/3501)	559	left for compatibility	29
2004-01-28 ltvers.dtx v1.0k		2005-11-08 ltoutenc.dtx v1.99h	
General: Check for old format		General: Added <code>\ij</code> and <code>\IJ</code> from	
made 5 years (pr/3601)	33	babel. (pr/3771) . .	115, 119, 121
2004-02-02 fontdef.dtx v2.3		2005-11-10 ltmath.dtx v1.1g	
General: Many things from here on		<code>\l</code> : (MH) Fixed potential problem	
made robust	278	in <code>\l</code> (pr/3399).	354
2004-02-02 ltoutenc.dtx v1.99		General: (MH) Minor	
General: Added <code>\textbigcircle</code>	125	documentation fixes.	348
2004-02-04 fontdef.dtx v2.3a		2006-05-18 ltboxes.dtx v1.1g	
General: Added <code>bigtriangle</code>		<code>\@parboxto</code> : Ensure <code>\@parboxto</code>	
synonyms for <code>stmaryrd</code>	276	holds the value of <code>\@tempdimb</code>	
2004-02-04 ltspace.dtx v1.3		not the register itself	
<code>\nbreakdashes</code> : (Macro added .	84	(pr/3867)	383
2004-02-06 ltoutenc.dtx v1.99d		2006-09-13 ltoutput.dtx v1.1m	
<code>\@inmathwarn</code> : New command		General: Ensure that rule is in	
added to fix severe bug:		<code>\normalcolor</code>	545
pr/3563	106		

2007-08-05 ltxclass.dtx v1.1h	2014-04-24 ltoutput.dtx v1.2n
\@fileswithoptions: Prevent loss of brackets PR/3965	\fl@tracemessage: Renamed internal trace commands; provide as package
2007-08-06 ltcentrl.dtx v1.0h	2014-04-27 ltfloat.dtx v1.2b
\@fornoop: Really make defs long	\end@dblfloat: Inline the code to allow some coexistence with packages that hook into \end@float and do not know about the algorithm change .
2007-08-31 ltssdcl.dtx v3.0l	2014-06-10 ltfloat.dtx v1.2b
\SetSymbolFont@: Font warning changed to info for encoding change (pr/3975)	\end@dblfloat: missing \fi added
2009-09-24 ltvers.dtx v1.0l	2014-12-30 ltfinal.dtx v2.0a
General: Stop checking for old format	\newmarks: macro added
2009-10-20 ltssdcl.dtx v3.0m	\newXeTeXintercharclass: macro added
\in@: More robust thanks to Heiko.	2014-12-30 ltfloat.dtx v1.2a
2009-10-28 lttextcomp.dtx v1.99k	\@textsubscript: Command added (latexrelease)
General: Added Latin Modern and TeX Gyre subsets	\textsubscript: Command added (latexrelease)
2009-11-04 lttextcomp.dtx v1.99l	2014-12-30 ltssbas.dtx v3.0y
General: Added more Latin Modern and TeX Gyre subsets	\mathgroup: move allocation to ltxplain.
2009-12-14 ltfntcmd.dtx v3.4a	2014-12-30 ltoutput.dtx v1.2m
\ifmaybe@ic: Macro added	General: Command updated (latexrelease)
\maybe@ic@: Use switch \ifmaybe@ic instead of \if@tempswa	2014-12-30 ltxplain.dtx v2.0a
\@st@ic: Use switch \ifmaybe@ic instead of \if@tempswa	\@alloc: macro added
2010-08-17 ltmiscen.dtx v1.1k	\@alloc@chardef: macro added
\enddocument: Use braces around \input arg (pr/4124)	\@alloc@top: macro added
2010-08-17 ltmiscen.dtx v1.1l	\@ch@ck: macro added
\enddocument: Change of plan: use \@@input instead (pr/4124) .	\extrafloats: macro added
2011-05-08 ltssdcl.dtx v3.0n	\newlanguage: New engine-specific allocation scheme (latexrelease)
\in@: Simplified thanks to Bruno.	2014-12-30 ltxspace.dtx v1.3b
2011-08-19 ltxclass.dtx v1.1i	\@: \@ discards spaces when moving (pr3039)(latexrelease)
\@ifclasswith: Re-jig definition after more stringent \in@ test.	2015-01-03 ltdefns.dtx v1.4a
2011-09-03 ltssdcl.dtx v3.0o	\typein: use modified definition in luatex
\new@mathversion: (Will) Remove \global before \newcount (unnecessary and caused etex bug).	2015-01-03 ltxdirchk.dtx v1.1
2012-01-20 ltxplain.dtx v2.0b	General: Enable extra primitives when LuaTeX is used
\loggingall: etex tracing if available	2015-01-03 ltfinal.dtx v2.0a
2013-07-07 ltxclass.dtx v1.1i	General: Skip resetting codes with Unicode engines
General: Correctly describe how the date in \@ifpackagelater is used	Unicode data loading added . .
2014-04-18 ltoutput.dtx v1.1o	2015-01-07 ltvers.dtx v1.0n
General: Handle infinite glue from \enlargethispage (pr/4023)	\IncludeInRelease: macro added

2015-01-08 ltboxes.dtx v1.1h	(latexrelease)	79
<code>\framebox</code> : Make Robust	(latexrelease)	381
<code>\makebox</code> : Make Robust	(latexrelease)	377
<code>\parbox</code> : Make Robust	(latexrelease)	382
<code>\raisebox</code> : Make Robust	(latexrelease)	387
<code>\rule</code> : Make Robust	(latexrelease)	386
<code>\savebox</code> : Make Robust	(latexrelease)	379
2015-01-08 ltdefs.dtx v1.4a	<code>\MakeRobust</code> : Added macro	47
2015-01-08 ltlength.dtx v1.1c	<code>\setlength</code> : to ensure first length argument is terminated. (latexrelease)	153
2015-01-08 ltmath.dtx v1.1h	<code>\)</code> : Make Robust (latexrelease) . .	354
	<code>\]</code> : Make Robust (latexrelease) . .	354
2015-01-09 ltfssini.dtx v3.1a	<code>\em</code> : Allow <code>\emph</code> to produce small caps (latexrelease)	259
	<code>\eminershape</code> : macro added (latexrelease)	259
2015-01-09 ltspace.dtx v1.1h	<code>\addpenalty</code> : Donald Arseneau’s fix from PR/377703 (latexrelease)	81
2015-01-10 ltcounts.dtx v1.1h	<code>\fnssymbol</code> : Unse <code>\TextOrMath</code> (latexrelease)	151
	<code>\stpelit</code> : Reset all within counters in one go (latexrelease)	148
2015-01-11 ltcounts.dtx v1.1h	<code>\TextOrMath</code> : Add command to solve robustness issues (pr/3752) (latexrelease)	151
2015-01-11 ltfloat.dtx v1.2b	<code>\dblfloatplacement</code> : float order in 2-column (latexrelease) . .	459
	<code>\xfloat</code> : Check for valid option (latexrelease)	454
	<code>\enddblfloat</code> : float order in 2-column (latexrelease)	458
2015-01-11 ltfssbas.dtx v3.0y	<code>\DeclareMathSizes</code> : Allow arbitrary units (latexrelease)	160
2015-01-11 ltspace.dtx v1.3d	<code>\Esphack</code> : Allow hyphenation (Donald Arseneau pr/3498)	
	(latexrelease)	78
2015-01-14 ltoutput.dtx v1.2n	<code>\@addtocurcol</code> : float order in 2-column (latexrelease)	516
	<code>\@addtodblcol</code> : float order in 2-column (latexrelease)	527
	<code>\@addtonextcol</code> : float order in 2-column (latexrelease)	523
	<code>\doclearpage</code> : Empty kludgeins box if necessary, pr/3528 . . .	499
	float order in 2-column (latexrelease)	499
	<code>\startdblcolumn</code> : float order in 2-column (latexrelease)	511
	<code>\xtryfc</code> : float order in 2-column (latexrelease)	513
	<code>\ztryfc</code> : float order in 2-column (latexrelease)	514
2015-01-14 ltspace.dtx v1.3e	<code>\addpenalty</code> : Avoid adding redundant skips (DPC)	82
2015-01-17 ltvers.dtx v1.0m	<code>\IncludeInRelease</code> : modified with <code>\currname</code>	34
2015-01-19 ltvers.dtx v1.0o	<code>\IncludeInRelease</code> : Optional argument	34
2015-01-20 ltoutput.dtx v1.2m	<code>\fl@tracemessage</code> : Reset <code>\IncludeInRelease</code> flags . . .	536
2015-01-22 ltvers.dtx v1.0p	General: Preserve any <code>\everyjob</code> material inserted by a loader (.ini file)	34
2015-01-23 ltfinal.dtx v2.0b	<code>\newmarks</code> : use reserved count 256	612
	<code>\newXeTeXintercharclass</code> : use reserved count 257	612
2015-01-23 ltplain.dtx v2.0c	<code>\extrafloats</code> : reserve counts 256–265	20
2015-01-24 ltfinal.dtx v2.0c	General: Skip T1-code entirely with Unicode engines	614
2015-02-03 ltfinal.dtx v2.0d	General: Set <code>\lccode</code> for – with Unicode engines	615
2015-02-16 ltoutenc.dtx v1.99m	General: Added <code>\textcommabelow</code> latex/4414	117

2015-02-16 ltoutenc.dtx v1.99n	automatically part of the kernel	218
General: Added <code>\textcommaabove</code>		117
Added composites for <code>\c</code>		125
Added composites for <code>\c</code>		120
2015-02-16 lttextcomp.dtx v1.99m		
General: Added lmtt (Heiko Oberdiek) latex/4415		326
2015-02-19 ltvers.dtx v1.0q		
<code>\IncludeInRelease</code> : Swap argument order		34
2015-02-20 ltplain.dtx v2.0d		
<code>\loggingall</code> : Spell commands correctly :-)		30
2015-02-21 ltdefs.dtx v1.4b		
General: Removed autoload support		36
2015-02-21 lterror.dtx v1.2o		
General: Removed autoload support		60
2015-02-21 ltfiles.dtx v1.1m		
General: Removed autoload support		88
2015-02-21 ltffsbas.dtx v3.0z		
General: Removed autoload code		154
2015-02-21 ltffscmp.dtx v3.0d		
General: Removed autoload code		218
2015-02-21 ltffsdcl.dtx v3.0p		
General: Removed autoload code		222
2015-02-21 ltfsstrc.dtx v3.0k		
General: Removed autoload code		195
2015-02-21 ltoutenc.dtx v1.99m		
General: Removed autoload code		102
2015-02-21 ltoutput.dtx v1.2n		
General: Removed autoload code		478
<code>\f@depth</code> : macro added(latexrelease)		498
2015-02-21 ltpictur.dtx v1.1k		
General: Removed autoload code		411
2015-02-21 ltplain.dtx v2.0e		
General: Removed autoload code		15
2015-02-21 lttab.dtx v1.1n		
General: Removed autoload code		389
2015-02-21 ltvers.dtx v1.0r		
General: Removed autoload code		33
2015-02-21 ltvers.dtx v1.0w		
<code>\IncludeInRelease</code> : set <code>\@currname</code> empty here (in case <code>\IncludeInRelease</code> input early)		34
2015-02-22 ltffscmp.dtx v3.0e		
General: Moved all code into latexrelease - obsolete commands are no longer		
2015-03-02 ltplain.dtx v2.0f		
<code>\e@mathgroup@top</code> : macro added		19
<code>\newlanguage</code> : allow 255 math groups in Unicode engines		18
2015-03-10 ltplain.dtx v2.0g		
<code>\hideoutput</code> : macro added		31
<code>\loggingall</code> : Reorganise to be less noisy		30
<code>\tracingnone</code> : macro added		31
2015-03-12 ltoutput.dtx v1.2m		
General: initialise <code>\@dbldeferlist</code> again		490
2015-03-18 ltffsdcl.dtx v3.0q		
<code>\DeclareSymbolFont</code> : Restrict Symbol fonts to 0-15		230
<code>\document@select@group</code> : Introduce <code>\e@mathgroup@top</code>		226
<code>\select@group</code> : Introduce <code>\e@mathgroup@top</code>		224
2015-03-26 ltfinal.dtx v2.0d		
General: Use renamed <code>unicode-letters.def</code>		614
2015-04-07 ltffsbas.dtx v3.1a		
<code>\wrong@fontshape</code> : Try loading fd file if family has changed		170
2015-04-28 ltfinal.dtx v2.0f		
<code>\newXeTeXintercharclass</code> : define <code>\xe@alloc@intercharclass</code> for compatibility with older xelatex initalisation		612
2015-05-10 ltlists.dtx v1.0t		
<code>\@doendpe</code> : Explicitly reset <code>\clubpenalty</code> before clearing <code>\everypar</code> ; see also pr/0462 and pr/4065		371
2015-06-19 ltfinal.dtx v2.0g		
<code>\e@alloc@intercharclass@top</code> : Use <code>-1</code> for first range to get contiguous allocation		612
<code>\newmarks</code> : Use <code>-1</code> for first range to get contiguous allocation		612
2015-06-19 ltplain.dtx v2.0h		
General: delete spurious old definition of <code>\newtoks</code>		23
<code>\e@alloc</code> : extra braces in case arguments not single token		20
<code>\newlanguage</code> : Use <code>-1</code> for first range to get contiguous allocation		18
2015-06-23 ltfinal.dtx v2.0h		
General: set <code>\patch@level</code> in ltvers rather than in		

ltfinal/ltpatch	624	provides_module: Function	
2015-06-23 ltvers.dtx v1.0t		added	596
General: set \patch@level in		luatexbase: Table added	596
ltvers rather than in		2015-10-02 ldirchk.dtx v1.2a	
ltfinal/ltpatch	33	General: Allow backing out of	
2015-08-06 ltplain.dtx v2.0i		unprefixed names	3
\extrafloats: Add \string in		2015-10-02 ltuatex.dtx v1.0b	
case argument is not an		General: Fix backing out of T _E X	
unexpandable primitive	21	code	595
2015-08-23 ldirchk.dtx v1.2		2015-10-02 ltuatex.dtx v1.0c	
General: Do not use luatex prefix	3	General: Allow backing out of Lua	
2015-08-23 ltvers.dtx v1.0v		code	595
General: Allow negative patchlevel		2015-10-02 ltuatex.dtx v1.0e	
for pre-release	34	uninstall: Function added	610
2015-08-30 ltplain.dtx v2.1a		2015-10-03 ltuatex.dtx v1.0f	
\newinsert: new \newinsert		provides_module: use	
implementation	22	luatexbase.log	596
2015-09-205 ltoutput.dtx v1.3a		2015-10-27 ltplain.dtx v2.1b	
General: extended \@freelist	489	\extrafloats: Use global	
2015-09-24 ltuatex.dtx v1.0a		assignment when switching to	
call_callback: Function added	607	extended range	20
callback_register: Function		2015-11-07 ltspace.dtx v1.3f	
modified	603	\@esphack: Only space if there is	
callback_descriptions: Function		no space at the end of the hlist	
added	609	latex/4443	78
\catcodetable@atletter: Macro		2015-11-14 ltuatex.dtx v1.0g	
added	592	General: Track LuaT _E X changes	
\catcodetable@initex: Macro		for (new)token.create	598
added	592	2015-11-18 ltplain.dtx v2.2a	
\catcodetable@latex: Macro		\newlanguage: Extended stream	
added	592	allocation in luatex (0.85)	18
\catcodetable@string: Macro		2015-11-19 ltplain.dtx v2.2b	
added	592	\newlanguage: Only extend	
add_to_callback: Function		allocation of write streams (see	
added	607	luatex list)	18
remove_from_callback: Function		2015-11-27 ltuatex.dtx v1.0h	
added	608	callback_descriptions: Match	
new_attribute: Function added	599	test in in-callback latex/4445	609
disable_callback: Function		in_callback: Guard against	
added	609	undefined list latex/4445	609
in_callback: Function added	609	2015-11-29 ltuatex.dtx v1.0i	
\newattribute: Macro added	591	General: Declare this as local	
\newcatcodetable: Macro added	591	before used in the module error	
\newluabytecode: Macro added	594	definitions (PHG)	596
\newluachunkname: Macro added	594	call_callback: Check name is not	
\newluafunction: Macro added	593	nil in error message (PHG)	607
\newwhatsit: Macro added	594	create_callback: Check name is	
module_error: Function added	597	not nil in error message	
module_info: Function added	597	(PHG)	606
module_warning: Function added	597	2015-12-02 ltuatex.dtx v1.0j	
modules: Function modified	596	General: Adjust hashtokens to	
create_callback: Function		store the result of	
added	606	tex.hashtokens()), not the	
		function (PHG)	598

Assorted typos fixed (PHG) . . .	589	process_rule, [hv]pack_quality	
Declaration/use of first_head fixed (PHG)	597	append_to_vlist_filter added	602
Remove nonlocal iteration variables (PHG)	589	read_cidmap_file added	602
Remove unreachable code after calls to error() (PHG)	589	show_warning_message added	603
2015-12-02 ltuatex.dtx v1.0k		token_filter removed	602
General: resolve name and i.description (PHG)	605	2016-02-18 ltfsdcl.dtx v3.0r	
call_callback: Give more specific error messages (PHG)	607	\@DeclareMathDelimiter: Check for delimiter not \delimiter	240
add_to_callback: Give more specific error messages (PHG)	607	\DeclareMathAccent: Check for mathaccent not \mathaccent	236
remove_from_callback: adjust initialisation of cb local (PHG)	608	\DeclareMathRadical: Check for radical not \radical	243
Give more specific error messages (PHG)	608	\DeclareMathSymbol: Check for mathchar not \mathchar	238
create_callback: Give more specific error messages (PHG)	606	2016-03-13 ltuatex.dtx v1.0n	
2015-12-10 ltfinal.dtx v2.0i		General: contribute_filter added	602
General: Use new common Unicode data loaders	614	insert_local_par added	602
2015-12-18 ltuatex.dtx v1.0l		2016-03-29 ltpictur.dtx v1.1l	
General: Load Unicode data from source	592	\@oval: add setting of line tests	428
2016-01-04 ltfinal.dtx v2.0j		initialise tests	428
General: Do not set up inter character classes for XeTeX	614	\@ovhorz: use glue not leaders if horizontal line not required	428
\e@alloc@intercharclass@top: Start allocation at one not three	612	\@ovvert: use glue not leaders if vertical line not required	428
2016-01-05 ltfinal.dtx v2.0k		\if@ovhline: macro added (latex/4452)	427
\e@alloc@intercharclass@top: Remove duplicated code	612	\if@ovvline: macro added (latex/4452)	427
2016-01-05 ltfinal.dtx v2.0l		2016-04-22 ltfinal.dtx v2.0q	
General: Correct latexrelease guards	614	\e@alloc@intercharclass@top: XeTeX 0.99996 has 4096 char classes not 256	612
Ensure old definitions for inter-character class toks are available using latexrelease	614	2016-06-19 ltoutenc.dtx v1.99m	
Missing brace	614	General: OT1 definition (was duplicate T1 definition)	120
2016-01-05 ltfinal.dtx v2.0m		2016-06-20 ltclass.dtx v1.1j	
General: Undefine XeTeX classes when using patching an older kernel	614	\@ifclasslater: don't declare as \@onlypreamble	556
2016-01-05 ltfinal.dtx v2.0p		2016-07-29 ltplain.dtx v2.2c	
General: Only apply XeTeX change if XeTeX is in use	614	\extrafloats: use \global \chardef	21
2016-02-11 ltuatex.dtx v1.0m		\newinsert: fix for tlb-newinsert-001	22
General: pdf_stream_filter_callback removed	603	2016-10-02 ltclass.dtx v1.2a	
		\@ifclasswith: Ignore spaces while checking for option clash	557
		\ExecuteOptions: Ignore spaces in argument	561
		2016-10-15 ltdirchk.dtx v1.2b	
		General: Require eTeX	4
		2016-10-15 lterror.dtx v1.2p	
		General: Require eTeX	60
		2016-10-15 ltfinal.dtx v2.0r	
		General: Require eTeX	612

2016-10-15 ltfinal.dtx v2.0s		2017-02-18 ltluatex.dtx v1.1c	
General: Tidy up status of char		new_attribute : Parameterise	
127	612	count used in tracking	599
2016-10-15 ltfssini.dtx v3.1b		new_bytecode : Parameterise count	
General: Require eTeX	247	used in tracking	600
2016-10-15 ltplain.dtx v2.2d		new_chunkname : Parameterise	
General: Require eTeX	15	count used in tracking	600
2016-10-16 ltplain.dtx v2.3a		new_whatst : Parameterise count	
\newlanguage : Allow languages up		used in tracking	599
to 16383 in luatex	18	2017-02-19 ltoutenc.dtx v2.0f	
2016-10-19 ltcounts.dtx v1.1j		General: add \@empty to guard	
\TextOrMath : Test directly for		against 3rd argument being	
\protected	151	empty	119
2016-11-06 ltplain.dtx v2.3b		declare composites with empty	
General: Drop \outer entirely . .	15	base for hat and tilde, use same	
2016-11-09 ltclass.dtx v2.1b		slots for \textasciicircum ans	
\@fileswithoptions : Improve		\textasciitilde	133
\ifx tests PR/4497	564	declare straight quotes using	
2016-11-17 ltluatex.dtx v1.0p		new \remove@tlig command	133
General: call_edit added	603	2017-02-22 ltoutenc.dtx v2.0g	
2016-12-03 fontdef.dtx v3.0a		General: Fix typo introduced at	
General: (DPC) Default to TU		2.0f	133
encoding for Unicode TeX		2017-02-24 ltoutenc.dtx v2.0h	
engines	266	General: introduce	
\shapedefault : (DPC) Default to		\DeclareUnicodeAccent	133
TU encoding for Unicode TeX		\DeclareTextCompositeCommand :	
engines	270	add check whether the accent	
2016-12-04 ltoutenc.dtx v2.0a		command is defined for this	
General: Added TU encoding	133	encoding	109
2017-01-01 ltoutput.dtx v1.3b		2017-03-08 ltclass.dtx v1.2c	
General: make fpmin negative so		\@ifclasslater : add	
ignored even if float height is		\@parse@version@dash to	
negative	544	support yyyy-mm-dd as well as	
2017-01-10 ltfssbas.dtx v3.2a		yyyy/mm/dd	556
\showhyphens : Add version of		2017-03-09 ltfinal.dtx v2.0t	
\showhyphens that works with		\l@nohyphenation : ensure	
XeTeX.	175	\l@nohyphenation is defined.	617
2017-01-23 ltoutenc.dtx v2.0b		2017-03-09 ltmiscen.dtx v1.1m	
General: Added TU specific		\@verbatim : Use \language not	
commands in ASCII range		\hyphenchar	344
pr/4500	133	\verb : Use \language to stop	
2017-01-24 ltoutenc.dtx v2.0c		hyphenation	346
General: Declare TU composites		2017-03-10 ltfiles.dtx v1.1n	
for i and j	133	\document : Save language default	91
Make \textasteriskcentered		2017-03-10 ltoutput.dtx v1.3c	
U+2217 not U+204E	133	\@writsetup : Reset \language	505
TeX ligature syntax for xetex		2017-03-13 ltdefs.dtx v1.5a	
and luatex reversed	133	\- : Define \- in terms of	
2017-01-24 ltoutenc.dtx v2.0d		\hyphenchar	51
General: Declare macron		2017-03-27 ltdefs.dtx v1.5b	
composites for YyGg	133	\@dischyph : Define \@dischyph	
2017-02-12 ltoutenc.dtx v2.0e		after \-	51
General: Declare fallback code for			
\textasteriskcentered	133		

2017-03-28 ltuatex.dtx v1.1e	2018-04-07 ltfinal.dtx v2.1c
General: glyph_stream_provider	\UseRawInputEncoding: Undefine
added 603	\inputencodingname 619
2017-03-29 ltboxes.dtx v1.3a	2018-04-08 ltclass.dtx v1.2g
\@arrayparboxrestore: Reset	\@ifclasslater: Strip leading
\lineskiplimit 384	spaces from dates. 556
2017-04-05 ltoutenc.dtx v2.0i	2018-04-08 ltclass.dtx v1.2h
\DeclareTextCompositeCommand:	\@onefilewithoptions: Pass
Declare accent command if not	expanded date 576
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 619
\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) 149
\newpage: account for the depth of	2018-05-02 ltuatex.dtx v1.1g
the last row of the page 492	General: find_sfd_file removed . . . 602
2017-12-17 ltoutput.dtx v1.4b	finish_synctex_callback added . . 603
\@addtonextcol: fix doc guards . 523	glyph_not_found added 603
2018-01-06 ltdefs.dtx 1.5c	read_sfd_file removed 602
\@ifundefined: Avoid defining	2018-05-08 ltclass.dtx v1.2i
undefined commands to \relax 49	\pkgcls@parse@date@arg: Make
2018-02-18 ltclass.dtx v1.2d	suspicious rollback a warning
\@ifclasslater: Added 0 up front	not error: github issue 43 . . 577
to make bad data come out as	2018-05-11 ltfinal.dtx v2.18
0. 556	General: Make invalid UTF-8 also
General: Introduce rollback	safe, for legacy filesystem
concept 574	encodings 619
2018-03-08 ltcounts.dtx v1.1k	2018-05-29 ltclass.dtx v1.2j
\@ifbothcounters: Interface	\endfilecontents: use \csname
added 149	not \@undefined 572
\@removefromreset: Interface	2018-08-11 ltoutenc.dtx v2.0j
added 148	General: Provide \guillemetleft
\counterwithin: Interface added 149	and
\counterwithout: Interface	\guillemetright . 122, 127, 135
added 149	2018-08-18 ltuatex.dtx v1.1h
2018-03-24 ltclass.dtx v1.2e	General: append_to_vlist_filter is
\pkgcls@use@this@release: Use	exclusive 602
full file name for old release . 580	2018-08-24 ltfinal.dtx v2.1f
2018-03-25 ltfinal.dtx v2.1a	\document@default@language:
General: default to UTF-8 618	Add to latexrelease
\UseRawInputEncoding: Macro	(github/68) 617
added 619	2018-09-02 ltsect.dtx v1.1b
2018-03-27 ltclass.dtx v1.2f	\@dottedtocline: Prevent
\endfilecontents: Use full file	protrusion
name for old release 571	(https://tex.stackexchange.com/q/172785/10109) 448
2018-04-06 ltfinal.dtx v2.1b	2018-09-24 fontdef.dtx v3.0b
\UseRawInputEncoding: Undo	General: Start LR-mode if
changes to	necessary (git/49) 282
\DeclareFontEncoding@ and	2018-09-24 ltmath.dtx v1.2b
definition of	\smash: Start LR-mode if
\DeclareUnicodeCharacter . 619	necessary (git/49) 351

<code>\vphantom</code> : Start LR-mode if necessary (git/49)	351	it is usable in normal text (github/70)	344
2018-09-24 ltspace.dtx v1.3h		Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	344
<code>\enspace</code> : Start LR-mode if necessary (git/49)	85		
<code>\leavevmode@ifvmode</code> : Macro added (git/49)	85	2018-10-21 ltuatex.dtx v1.1i	
2018-09-26 ltdefs.dtx v1.5e		<code>new_luafunction</code> : Function added	601
<code>\renew@command</code> : Always explicitly generate a space after the csname and not rely on <code>\noexpand</code> to save tokens (git/41)	41	2018-11-09 ltbibl.dtx LaTeX2e	
2018-09-26 ltmiscen.dtx v1.1n		<code>\bibliography</code> : Zap spaces in the argument as BibTeX doesn't support them (github/88)	473
<code>\@writefile</code> : Sometimes mask the newline char when writing to files (github/73)	337	2018-11-18 ltoutenc.dtx v2.0k	
<code>\add@percent@to@temptokena</code> : Sometimes mask the newline char when writing to files (github/73)	336	General: Provide <code>\Hwithstroke</code> and <code>\hwithstroke</code>	137
<code>\protected@file@percent</code> : Sometimes mask the newline char when writing to files (github/73)	336	2018-11-19 ltoutenc.dtx v2.0k	
2018-09-26 ltsect.dtx v1.1c		General: Added <code>\Hwithstroke</code> and <code>\hwithstroke</code>	121
<code>\addcontentsline</code> : Sometimes mask the newline char when writing to files (github/73)	447	2018-11-28 ltoutput.dtx v1.4d	
2018-10-10 ltspace.dtx v1.3i		<code>\@combinedblfloats</code> : Unbox <code>\@outputbox</code> to preserve boxing level (github/94)	510
<code>\@esphack</code> : Don't introduce breakpoints if <code>@nobreak</code> is true and after sections	78	2018-12-30 lttab.dtx v1.1p	
2018-10-11 ltmiscen.dtx v1.1o		<code>\@tabclassz</code> : Add extra <code>\hskip</code> to guard against an <code>\unskip</code> at the start of a c-column cell (gh/102)	407
<code>\@setupverbvisibleospace</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	345	2019-02-07 ltfiles.dtx v1.1o	
<code>\@sverb</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	345	<code>\@swaptwoargs</code> : Helper macro added	98
<code>\@verbvisibleospacebox</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	345	<code>\InputIfFileExists</code> : Expand <code>\@filef@und</code> before executing second argument (github/109)	98
<code>\asciispace</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	344	2019-02-07 ltfssbas.dtx v3.2b	
<code>verbatim*</code> : Provide visible space in <code>\verb*</code> also for XeTeX and LuaTeX (github/69)	346	<code>\define@newfont</code> : Changed wording of warning (github/107)	166
<code>\verbvisibleospace</code> : Provide <code>\verbvisibleospace</code> such that		2019-06-18 ltuatex.dtx v1.1j	
		General: <code>finish_synctex_callback</code> renamed <code>finish_synctex</code>	603
		<code>font_descriptor_objnum_provider</code> added	603
		<code>make_extensible</code> added	603
		<code>new_graf</code> added	602
		<code>page_objnum_provider</code> added	603
		<code>process_pdf_image_content</code> added	603
		<code>wrapup_run</code> added	603
		2019-07-01 ltfiles.dtx v1.2a	
		<code>\IfFileExists</code> : Support UTF-8	97
		<code>\includeonly</code> : Support UTF-8	94
		<code>\set@curr@file</code> : Support UTF-8	96

2019-07-09 ltffsbas.dtx v3.2c	2019-08-27 ltmiscen.dtx v1.1p
<code>\DeclareErrorFont</code> : Don't set any	General: Make various commands
<code>\f@...</code> macros 169	robust 333
2019-07-09 ltffsini.dtx v3.1c	<code>\begin</code> : Make command robust . . . 338
General: Explicitly set some	<code>\end</code> : Make command robust . . . 340
defaults 262	2019-08-27 ltoutput.dtx v1.4e
2019-08-22 ltxref.dtx v1.1l	<code>\@begindvibox</code> : Make
<code>\labelformat</code> : Commanded moved	<code>\AtBeginDvi</code> robust 490
from <code>varioref.sty</code> 332	2019-08-27 ltpage.dtx v1.0l
<code>\Ref</code> : Commanded moved from	General: Make various commands
<code>varioref.sty</code> 332	robust 475
<code>\refstepcounter</code> : Allow <code>\p@...</code>	2019-08-27 ltpictur.dtx v1.1m
to have an argument 332	General: Make various commands
2019-08-27 fontdef.dtx v3.0c	robust 411
General: Various commands made	Remove several unnecessary
robust throughout the file .. 275	<code>\gdef</code> definitions 411
2019-08-27 ltboxes.dtx v1.3b	2019-08-27 ltsect.dtx v1.1d
General: Various commands made	General: Make various commands
robust 376	robust 438
2019-08-27 ltdefs.dtx v1.5f	2019-08-27 ltspace.dtx v1.3j
General: Make various commands	General: Make various commands
robust 52	robust 71
<code>\MakeRobust</code> : Make the	2019-08-27 lttab.dtx v1.1q
assignments global as we may	General: Make various commands
need to apply them inside a	robust 389
group 47	Remove several unnecessary
2019-08-27 ltfiles.dtx v1.2b	<code>\gdef</code> definitions 389
<code>\IfFileExists</code> : Make command	2019-08-30 lterror.dtx v1.2q
robust 97	<code>\conditionally@traceoff</code> : Macro
<code>\InputIfFileExists</code> : Make	added 68
command robust 98	<code>\conditionally@traceon</code> : Macro
2019-08-27 ltffsbas.dtx v3.2d	added 68
General: Make various commands	2019-09-09 ltffsdcl.dtx v3.0s
robust 154	<code>\DeclareMathSymbol</code> : Allow
2019-08-27 ltffsdcl.dtx v3.0s	definition if the math symbol
<code>\DeclareMathAccent</code> : Make math	was a command already
accents robust 236	robust 238
<code>\set@mathdelimiter</code> : Make math	2019-09-14 ltfinal.dtx v2.1h
delimiters robust 242	<code>\uc1clist</code> : Expand UTF8 chrs
2019-08-27 ltffsini.dtx v3.1d	when case changing
General: Make various commands	(github/177) 623
robust 247	2019-09-16 ltxref.dtx v1.1m
2019-08-27 ltidxglo.dtx v1.1f	General: Correctly revert the
General: Make <code>\index</code> and	<code>\p@...</code> change 332
<code>\gloassary</code> robust 469	2019-09-21 fontdef.dtx v3.0d
2019-08-27 ltlength.dtx v1.1d	General: Distangle alias
General: Make various command	(gh/184) 277, 281
robust 153	2019-10-02 ltfinal.dtx v2.2
2019-08-27 ltlogos.dtx v1.1j	General: Load <code>ltxexpl</code> 625
<code>\TeX</code> : Make <code>\TeX</code> command robust 87	2019-10-02 ltluatex.dtx v1.1k
2019-08-27 ltmath.dtx v1.2c	General: <code>linebreak.filter</code> is
General: Make various commands	<code>exclusive</code> 602
robust 348	<code>mlist_to_hlist</code> is <code>exclusive</code> . . . 602
	<code>process_rule</code> is <code>exclusive</code> . . . 602

2019-10-07 lttab.dtx v1.1q		2019-12-17 fontdef.dtx v3.0e	
<code>\extracolsep</code> : This needs to		<code>\mddefault</code> : Set <code>\bfdefault</code> to	
expand	400	“b”	269
2019-10-11 ltfiles.dtx v1.2c		<code>\updefault</code> : Set <code>\updefault</code> to	
<code>\set@curr@file</code> : Remove one		“up”	269
brace group	96	2019-12-17 lftncmd.dtx v3.4c	
2019-10-11 ltfststrc.dtx v3.0l		<code>textssc</code> : Macro added	291
<code>\@font@aliasinfo</code> : Added ‘alias’		2019-12-17 ltffsbas.dtx v3.2e	
size function	216	<code>\usefont</code> : Don’t call <code>\fontseries</code>	
2019-10-18 ltclass.dtx v1.3d		or <code>\fontshape</code>	162
<code>\@fileswithoptions</code> : Initialize		2019-12-17 ltffssini.dtx v3.1e	
<code>\dots-h@@k</code> only when loading		General: Provide custom series	
the package or class (gh/198)	565	settings a la mweights	248
2019-10-22 ltuatex.dtx v1.1j		<code>\DeclareEmphSequence</code> : Provide	
General: page_objnum_provider		<code>\emph</code> sequences	257
and process_pdf_image_content		2019-12-18 ltoutenc.dtx v2.0m	
classified data	603	General: Don’t fake	
2019-10-25 ltmiscen.dtx v1.1q		<code>\textcompwordmark</code> ; take	
<code>\add@percent@to@temptokena</code> :		default from T1 instead	116
Allow unbalanced conditionals		<code>\add@accent</code> : Avoid code that	
in #1 (gh/202)	336	breaks <code>\accent</code>	109
2019-10-26 ltfiles.dtx v1.2d		2019-12-21 fontdef.dtx v3.0e	
<code>\@iffileonpath</code> : quote on openin	97	General: Distangle alias	
<code>\IfFileExists</code> : dont quote name	97	(gh/184)	275–278
<code>\IfFileExists@</code> : quote on openin	97	2020-01-11 ltffssini.dtx v3.1f	
<code>\set@curr@file</code> : remove quotes .	96	<code>\rmfamily</code> : Streamlined	
2019-11-01 ltdirchk.dtx v1.3a		implementation with hook	252
<code>\filename@parse</code> : take last . not		<code>\ttfamily</code> : Streamlined	
first	12	implementation with	
2019-11-02 ltmiscen.dtx v1.1s		hook	252, 253
<code>\@centercr</code> : Make <code>\@centercr</code>		2020-01-20 ltffssdcl.dtx v3.0t	
robust (gh/203)	341	<code>\set@mathdelimiter</code> : fix for	
2019-11-02 ltspc.dtx v1.3k		gh/251	242
<code>\@normalcr</code> : Make also		2020-01-20 ltoutenc.dtx v2.0n	
<code>\@normalcr</code> robust	75	General: fix for gh/251	116
2019-11-09 ltfiles.dtx v1.2e		2020-01-22 lttextcomp.dtx v1.0b	
<code>\set@curr@file</code> : expand and		<code>\tc@subst</code> : The overall default is	
<code>\string</code> before removing		<code>\textcompsubstdefault</code> not	
quotes	96	<code>\substdefault</code>	300
2019-11-10 ltmiscen.dtx v1.1r		2020-01-25 fontdef.dtx v3.0f	
<code>\add@percent@to@temptokena</code> : fix		General: Load t1enc.def last	
to special comment catcodes		(gh/255)	266
(gh/202)	336	2020-01-25 ltoutenc.dtx v2.0m	
2019-11-11 ltfiles.dtx v1.2f		General: Load each encoding file	
<code>\@iffileonpath</code> : make		only once (gh/255)	143
<code>\@filef@und</code> match quoting		2020-01-27 ltclass.dtx v1.3g	
used on <code>\openin</code>	97	<code>\endfilecontents</code> : Fix typo in	
2019-11-22 ltoutenc.dtx v2.0l		error message	570
General: Avoid spurious if fontenc		2020-01-28 ltclass.dtx v1.3h	
selects LY1 as default encoding		<code>\endfilecontents</code> : Allow spaces	
(gh/199)	144	in option string and display	
2019-11-29 ltclass.dtx v1.3e		only unknown options not the	
<code>\@providepackage</code> : Protect		whole option list (gh/256)	569
package info text (gh/52)	558		

2020-01-31 ltvrs.dtx v1.1e		2020-02-11 ltssini.dtx v3.1h	
General: Allow for upcoming		General: Provide default value for	
format as pre-release 0	34	\@fontenc@load@list	
2020-02-02 ltluatex.dtx v1.1l		(gh/273)	262
General: Add reverselist callback		2020-02-11 ltoutenc.dtx v2.0o	
type	605	General: Update	
glyph_info added	603	\@fontenc@load@list with	
page_order_index added	603	option list (gh/273)	145
post_linebreak_filter is		2020-02-18 ltssini.dtx v3.1i	
reverselist	602	\bfseries: Make the \ifx	
create_callback: Provide proper		selection outside of	
fallbacks for user-defined		\fontseries argument so that	
callbacks without		it is not done several times	250
user-provided default handler	606	\mdseries: Make the \ifx	
2020-02-05 ltssini.dtx v3.1g		selection outside of	
\DeclareFontSeriesDefault:		\fontseries argument so that	
Clarrified error text	249	it is not done several times	250
Corrected misspelled csname		\prepare@family@series@update:	
(gh/264)	249	No series auto-update when	
2020-02-05 lttextcomp.dtx v2.0n		forced (gh/277)	253
General: Changed the package		Recognize current family if it is	
default to info (gh/262)	316	not a “meta” family and	
Ensure we are on a new format		auto-update series using	
(gh/260)	315	\bfdefault (gh/277)	254
2020-02-10 ltssaxes.dtx v1.0c		2020-02-20 ltclass.dtx v1.3j	
\fontseries: Switch		\endfilecontents: Fix missing	
\if@forced@series added	187	quotes around file name	
\fontseriesforce: Switch		(gh/284)	570
\if@forced@series added	187	2020-02-24 ltssbas.dtx v3.2f	
\if@forced@series: Switch		\DeclareFontShape@: Drop	
\if@forced@series added	187	surplus “m” in series when	
2020-02-10 ltssini.dtx v3.1h		defining fontshape (gh/289)	155
\@defaultfamilyhook: Add		2020-02-27 ltdefns.dtx v1.5g	
\@defaultfamilyhook to		\@gobblethree: Macro added	44
\normalfont (gh/269)	261	2020-02-27 ltssaxes.dtx v1.0d	
\reset@font: Add		\series@maybe@drop@one@m: Drop	
\@defaultfamilyhook to		“m” in certain values from a	
\normalfont (gh/269)	261	fixed list (gh/293)	189
2020-02-10 lttextcomp.dtx v1.0c		\set@target@series: Drop “m”	
General: Use \@tabacckludge for		only in a specific set of values	
tabbing where necessary		(gh/293)	188
(gh/271)	303	2020-02-27 ltssbas.dtx v3.2g	
2020-02-11 fontdef.dtx v3.0g		\DeclareFontShape@: Only “m” if	
General: Provide value for		the series value is a member of	
\@fontenc@load@list		a fixed list and issue warning if	
(gh/273)	266, 267	doing it (gh/293)	155

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
$\backslash"$	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, o387, x185, x202, Q420
$\backslash\#$	a62, a75, b6, b14, b420, d388, o374, Q403
$\backslash\$$ a74, b4, b13, d387, l307, l441, l448, l562, l798, l805, x547, x554, Q404	
$\backslash\%$	a75, a105, a107, a127, b14, b418, d388, l488, l490, o376, x556, x670, A56, N792, N793, Q405
$\backslash\&$ a74, b5, b13, b419, d387, N193, Q406	
\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, o386, t418, u247, x180, A351, B208, D245, D266, E60, M584, Q421
$\backslash($	<u>B225</u> , B299
$\backslash)$	b440, <u>B225</u> , B300
$\backslash*$	o379, <u>B205</u> , N603, N718, N794
$\backslash+$	E60
$\backslash,$	b354, b356, <u>i335</u> , u500, A351, B7, B8, B40, B154, B156, B159, B184, B201
$\backslash-$..	b322, d9, <u>d395</u> , i326, l413, l414, l568, l782, l783, o381, A351, D244, D265, E60, Q157, Q197
$\backslash.$	b353, b355, k43, k101, l228, l380, l449, l450, l471, l581, l582, l608, l609, l635, l742, l812, l819, l1185, l1267, l1268, l1277, l1278, l1287, l1288, l1305, l1366, l1367, l1397, l1398, o380, x186, x203
$\backslash/$	a97, d10, o328, o382, N192
$\backslash:$	b354, b356, d382, d383, <u>B206</u>
$\backslash;$	b354, b356, u494, B185, <u>B201</u>
$\backslash<$...	l569, l733, o377, A351, E59, E97
$\backslash=$	l229, l381, l470, l743, l1183, l1257, l1258, l1273, l1274, l1296, l1297, l1298, l1323, l1324, l1349, l1350, l1379, l1380, l1395, l1396, l1403, l1404, t418, x189, D245, D266, E59
$\backslash>$	l566, l734, o378, A351, <u>B201</u> , B206, E59
$\backslash?$	b353, b355, Q421
$\backslash@$	a65, d393, d394, g19, <u>i338</u> , j2, N24, N32, P18, P852, Q413
$\backslash@@$	a331, a332, f15, f19, f20, f21, f22, f24, f27, f28, f30, f31, k389, k405, q464, q466, q467, E214, E215, E216, E226, M10, M11
$\backslash@@defaultsubs$	<u>o556</u>
$\backslash@@enc@update$	l183, o259, <u>o263</u>
$\backslash@@end$	a69, a222, d8, k369, k370, A39, A86, O18, Q500, Q521
$\backslash@@endpbox$	E181, E212, <u>E362</u>
$\backslash@@eqncr$	B334, B352, <u>B355</u> , B460
$\backslash@@fileswith@opti@ns$	N266, <u>N489</u>
$\backslash@@hyph$	d9
$\backslash@@hyphenation$	<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, N945
$\backslash@@italiccorr$	<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>
$\backslash@@protect$	d229, d235, d244
$\backslash@@startpbox$	E181, E212, <u>E362</u>
$\backslash@@underline$	<u>D358</u> , D361, D362
$\backslash@@unprocessedoptions$...	N471, <u>N533</u>

File Key: a=ltmdirchk.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=ltfssaxes.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

- \@@warning [g166](#)
- \@Alpha [m106](#), [m122](#)
- \@DeclareEncodingSubset
 . [x66](#), [x68](#), [x69](#), [x70](#), [x71](#), [x74](#), [x81](#)
- \@DeclareMathDelimiter [s752](#), [s771](#)
- \@DeclareMathSizes .. [o206](#), [o207](#), [o209](#)
- \@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](#), [o625](#), [o632](#),
 [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 [t418](#), [D245](#), [D266](#)
- \@accii [t418](#), [D245](#), [D266](#)
- \@acciii [t418](#), [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](#), [t394](#), [t397](#), [t404](#),
 [t407](#), [t414](#), [t417](#), [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 [d194](#),
 [g185](#), [g187](#), [u260](#), [N666](#), [N756](#)
- \@badcrrr [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](#)
- \@begindocumenthook [k55](#),
 [k58](#), [k110](#), [k113](#), [K33](#), [N500](#), [N514](#)
- \@beginndvi [M623](#), [M682](#), [M710](#)
- \@beginndvibox [M86](#), [M711](#)
- \@beginparpenalty
 [i14](#), [B387](#), [B399](#), [B425](#), [C23](#), [C170](#)
- \@begintheorem [G30](#), [G35](#)
- \@bezier [F369](#), [F370](#)
- \@bibitem [K3](#), [K8](#)
- \@biblabel [K4](#), [K54](#)
- \@bitor [M15](#), [M881](#),
 [M901](#), [M937](#), [M960](#), [M1027](#),
 [M1111](#), [M1121](#), [M1269](#), [M1280](#),
 [M1422](#), [M1509](#), [M1627](#), [M1752](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- \@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](#)
- \@capytype M1995,
I5, I9, I12, I40, I88, I109, I157
- \@car 37, [d38](#), j14, l89, l110
- \@carcube [d40](#), d110
- \@cclv [b16](#), M300, M304, M382,
M383, M412, M429, M430,
M459, M483, M487, M488, Q53
- \@cclvi [b21](#), b57, b82, b93, b95, b99,
b159, b173, N664, N754, P28, P56
- \@cdr 37, [d38](#), d363, d364
- \@centercr [A153](#), A179, A183, A187
- \@centering B314, B315,
B322, B325, B328, B453, B457
- \@cflb [M714](#)
- \@cflt [M714](#)
- \@changed@cmd
. [l3](#), l63, l223, o131, o267, 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, N1026
- \@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, N879, N954, N976, N1006
- \@clsexextension [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](#)
- \@comdblflleft [M750](#)
- \@comfleleft M720, M736, [M750](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- M1934, M2235, M2275, [I280](#)
- [\dblflset](#) [I26](#)
- [\dblflpbot](#) [M2319](#), [I290](#), [I304](#)
- [\dblflpsep](#) [M2319](#), [I289](#), [I303](#)
- [\dblflptop](#) [M2319](#), [I288](#), [I302](#)
- [\dblftoplist](#) [M69](#), [M232](#),
[M235](#), [M237](#), [M397](#), [M398](#),
[M445](#), [M446](#), [M755](#), [M759](#),
[M761](#), [M762](#), [M1639](#), [M1644](#),
[M1764](#), [M1770](#), [M1925](#), [M1952](#)
- [\dblftopnum](#) [M107](#), [M127](#),
[M238](#), [M240](#), [M766](#), [M1578](#),
[M1579](#), [M1643](#), [M1646](#), [M1654](#),
[M1674](#), [M1679](#), [M1699](#), [M1700](#),
[M1769](#), [M1773](#), [M1781](#), [M1802](#),
[M1807](#), [M1918](#), [M1945](#), [I283](#), [I297](#)
- [\dblftoproom](#) [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](#)
- [\declareoptions](#)
[N8](#), [N209](#), [N232](#), [N248](#), [N263](#), [N498](#)
- [\declareoption](#) ... [N207](#), [N208](#), [N216](#)
- [\defaultfamilyhook](#) .. [t344](#), [t347](#), [t361](#)
- [\defaultsubs](#) .. [o510](#), [o544](#), [o556](#), [A26](#)
- [\defaultunits](#) [o214](#), [o218](#),
[o219](#), [o220](#), [o235](#), [o299](#), [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](#),
[u552](#), [u553](#), [u555](#), [u556](#), [D357](#),
[D401](#), [E175](#), [E207](#), [F107](#), [F158](#),
[F161](#), [F176](#), [F183](#), [F403](#), [M1851](#)
- [\dir](#) [a160](#), [a163](#), [a165](#), [a167](#), [a168](#)
- [\dischyph](#) [d395](#), [d410](#), [d415](#), [D244](#), [D265](#)
- [\doclearpage](#) [M296](#), [M371](#)
- [\@documentclasshook](#) . [N3](#), [N389](#), [N417](#)
- [\@doendpe](#) [A138](#), [C123](#)
- [\@dofilelist](#) [k395](#), [k411](#), [A21](#)
- [\@donoparitem](#) [C144](#), [C158](#)
- [\@dot](#) [F329](#), [F342](#)
- [\@dotsep](#) [H194](#), [H217](#)
- [\@dottedtocline](#) ... [H179](#), [H205](#), [H206](#)
- [\@downline](#) [F155](#), [F159](#), [F164](#)
- [\@downvector](#) [F126](#), [F164](#)
- [\@eha](#) [d253](#), [d280](#), [g170](#), [g188](#),
[g190](#), [g192](#), [g200](#), [g202](#), [g232](#),
[k158](#), [k173](#), [l52](#), [l84](#), [o28](#), [o58](#),
[o102](#), [o144](#), [o187](#), [o253](#), [o310](#),
[q106](#), [s25](#), [s70](#), [s99](#), [s161](#), [s192](#),
[s293](#), [s314](#), [s346](#), [s387](#), [s432](#),
[s437](#), [s492](#), [s610](#), [s614](#), [s618](#),
[s653](#), [s657](#), [s661](#), [s718](#), [s728](#),
[s813](#), [s818](#), [s821](#), [s853](#), [s856](#),
[s929](#), [s932](#), [s935](#), [s1002](#), [s1008](#),
[w146](#), [x24](#), [x102](#), [x824](#), [x833](#),
[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](#), [N1006](#)
- [\@ehd](#) . [g170](#), [g197](#), [g204](#), [g207](#), [g209](#),
[g215](#), [s118](#), [E88](#), [E97](#), [N351](#), [I6](#)
- [\@elt](#) [d37](#), [k210](#),
[l1508](#), [l1510](#), [m20](#), [m35](#), [m53](#),
[m56](#), [t119](#), [t130](#), [t132](#), [t133](#),
[t157](#), [t387](#), [u17](#), [u25](#), [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](#)
- [\@endparenv](#) [C120](#), [C123](#)
- [\@endparpenalty](#)
..... [i15](#), [B388](#), [B400](#), [B426](#), [C23](#), [C124](#)
- [\@endpbox](#) [E181](#),
..... [E212](#), [E242](#), [E309](#), [E360](#), [E363](#)

File Key: a=ltmdirchk.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=ltfssaxes.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=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- \@endpfalse ... [A100](#), [A117](#), [C129](#),
[C131](#), [C135](#), [C136](#), [C138](#), [D107](#)
- \@endpeltrue [C138](#)
- \@endpetrue [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](#)
- \@eqnum [B305](#), [B306](#), [B360](#), [B374](#), [B433](#)
- \@eqnset [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
.. [d192](#), [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](#),
[t394](#), [t404](#), [t414](#), [Q210](#), [Q526](#), [Q542](#)
- \@filesfalse [k128](#), [N588](#), [N589](#)
- \@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](#), [N629](#), [N633](#)
- \@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 [d187](#),
[k54](#), [k109](#), [k233](#), [l68](#), [l153](#),
[q300](#), [s53](#), [s81](#), [s142](#), [s172](#), [s763](#),
[A9](#), [B368](#), [E348](#), [K18](#), [K42](#),
[P65](#), [P100](#), [P108](#), [P166](#), [Q255](#), [I10](#)
- \@firstoftwo [a87](#), [d187](#),
[d337](#), [d347](#), [d357](#), [d384](#), [k263](#),
[k302](#), [l133](#), [m157](#), [m162](#), [s767](#),
[x90](#), [x745](#), [x796](#), [x812](#), [z19](#), [L16](#),
[N48](#), [N68](#), [N80](#), [N107](#), [N125](#), [N1015](#)
- \@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 [o133](#), [o171](#), [o177](#), [o337](#),
[o354](#), [o592](#), [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](#), [s601](#), [s644](#), [s709](#), [s803](#),
[s970](#), [s999](#), [x76](#), [x77](#), [x120](#), [Q374](#)
- \@font@shape@subst@warning
[p408](#), [p411](#), [p416](#), [p536](#), [p539](#), [p583](#)
- \@font@warning
.. [o3](#), [o506](#), [o511](#), [o538](#), [o545](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

- p419, q19, q33, q41, q49, q61,
q77, q430, q444, q456, q461,
q468, q511, q519, r30, A23, Q230
- `\@fontenc@load@list`
..... I1509, t387, u17, u25
- `\@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, N579
- `\@forced@seriesfalse` p387
- `\@forced@seriestrue` p386
- `\@forloop` f19, f20
- `\@fornoop` f15, f23, f29
- `\@fortmp` f17,
f18, f26, N281, N283, N578, N579
- `\@fpbot` M863, M2313, I290, I304
- `\@fpm` M113, M920, M1916, M1943,
M2165, M2182, I278, I287, I301
- `\@fps` M1993,
M1995, M1998, I41, I42, I44,
I47, I64, I110, I111, I113, I116, I133
- `\@fpsaddefault`
..... 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
- `\@getc` F223, F256, F300, F332
- `\@getf`psbit M980,
M1016, M1572, M1693, M2029
- `\@getl`arrow F124, F132, F134
- `\@getl`inechar F70, F109
- `\@getp`en i34, i37, i46, i91
- `\@getr`arrow F125, F132, F141
- `\@glossaryfile` J21, J22, J31
- `\@gnew`line i82, i84, i85
- `\@gobble` ... d86, d108, d183, d194,
d211, d215, d250, d256, d259,
d268, d277, d283, d286, d295,
d310, d314, d316, f6, f9, g101,
g127, g153, g162, i75, i381, k61,
k116, k236, k238, k385, l29,
o507, o540, 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,
s796, s806, s887, s892, s961,
s992, t117, t157, t397, t407,
t417, x773, A104, H143, H144,
H145, H146, H147, H176, K11,
K25, K26, M619, M620, M621,
M678, M679, M680, M927,
M1900, M2166, M2183, N339,
N557, N623, N652, N737, N742,
N816, N960, N972, P64, P98,
Q213, Q307, Q464, Q473, Q542, I7
- `\@gobble@I`ncludeInRelease
..... c86, c94, c103
- `\@gobblec`r i379, i380
- `\@gobblef`our d183,
s24, s252, s368, s370, s374, s376,
s386, s390, s514, s566, N654, N744
- `\@gobblet`hree d183
- `\@gobblet`wo . d150, d151, d183, f12,
k30, k88, o512, o546, s132, A16,
A24, L11, L13, N622, N736, Q236
- `\@gtemp`a 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
- `\@halign`to .. E158, E162, E165, E179
- `\@hangf`rom H66, H117, H138
- `\@height` b391, d11, i296,
i304, i293, i295, q144, u334,
u552, u553, u555, u556, D118,

File Key: a=ltmdirchk.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=ltfssaxes.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

- 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 [d190](#)
- \@if [d146](#), [d147](#), [d149](#)
- \@ifOpti@ns [N100](#),
[N104](#), [N106](#), [N123](#), [N124](#), [N140](#)
- \@if@ptions [N95](#), [N96](#), [N99](#), [N101](#), [N448](#)
- \@ifatmargin [E55](#), [E94](#)
- \@ifbothcounters
. [m61](#), [m69](#), [m71](#), [m79](#), [m81](#), [m93](#)
- \@ifclasslater [553](#), [N51](#)
- \@ifclassloaded [553](#), [N40](#)
- \@ifclasswith [553](#), [N95](#)
- \@ifdefinable [37](#), [d59](#),
[d61](#), [d105](#), [d107](#), [d213](#), [i14](#), [i17](#),
[m11](#), [n3](#), [t317](#), [D70](#), [G7](#), [G15](#), [G22](#)
- \@iffileopath [k248](#), [k256](#), [k286](#), [k295](#)
- \@ifl@aded [N40](#), [N41](#), [N44](#), [N50](#), [N447](#)
- \@ifl@t@r [i1474](#), [N56](#),
[N61](#), [N63](#), [N74](#), [N75](#), [N85](#), [N360](#)
- \@ifl@ter [i1513](#),
[i1514](#), [N51](#), [N52](#), [N55](#), [N58](#), [N476](#)
- \@ifl@ter@@ [i1513](#), [i1514](#)
- \@ifnch [d369](#), [d371](#), [d383](#)
- \@ifnextchar [37](#),
[a98](#), [d365](#), [d370](#), [d384](#), [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 [553](#), [N51](#)
- \@ifpackageloaded [553](#), [M1958](#), [N40](#)
- \@ifpackagewith [553](#), [N95](#)
- \@iframebox [D154](#), [D155](#), [D156](#)
- \@iframepicbox [D186](#), [D187](#)
- \@ifstar [37](#), [d48](#), [d384](#), [i59](#), [i71](#), [i280](#),
[i350](#), [m67](#), [m77](#), [o206](#), [r121](#),
[A158](#), [A165](#), [A338](#), [A347](#), [B343](#),
[E56](#), [E190](#), [E197](#), [F53](#), [F329](#),
[H52](#), [H142](#), [M1856](#), [N207](#), [N229](#)
- \@ifundefin@d@i [d327](#), [d328](#), [d345](#), [d348](#)
- \@ifundefin@d@ii [d327](#), [d330](#), [d333](#)
- \@ifundefined [37](#), [d102](#), [d109](#), [d129](#),
[d136](#), [d158](#), [d169](#), [d250](#), [d256](#),
[d277](#), [d283](#), [d310](#), [d322](#), [m3](#),
[m7](#), [m16](#), [m50](#), [m62](#), [m64](#), [o100](#),
[o186](#), [q378](#), [s287](#), [x75](#), [x775](#),
[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](#)
- \@iiparbox [D196](#), [D204](#),
[D211](#), [D214](#), [D215](#), [D216](#), [D321](#)
- \@iiminipage [D287](#), [D289](#)
- \@iinput [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](#), [d419](#)
- \@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 [i3](#)
- \@inpenc@test [Q252](#), [Q319](#)
- \@input [k32](#), [k90](#), [k181](#), [k357](#), [H152](#)
- \@input@ [k196](#), [k359](#), [o364](#), [K31](#)
- \@inputcheck
. [a70](#), [a191](#), [a192](#), [a195](#), [a203](#),
[d23](#), [d30](#), [k3](#), [k243](#), [k244](#), [k251](#),
[k260](#), [k261](#), [k264](#), [k281](#), [k282](#),
[k289](#), [k299](#), [k300](#), [k303](#), [N611](#),
[N612](#), [N648](#), [N724](#), [N725](#), [N732](#)
- \@insertfalse [M1073](#), [M1223](#),
[M1394](#), [M1475](#), [M1570](#), [M1691](#)
- \@inserttrue [M999](#), [M1044](#),
[M1161](#), [M1329](#), [M1649](#), [M1776](#)

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscnp.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

`\@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](#)
`\@iwhilesw` [f10](#)
`\@ixpt` [o688](#)
`\@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](#), [d251](#), [d278](#),
[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](#), [o28](#), [o58](#), [o102](#), [o144](#),
[o187](#), [o253](#), [o310](#), [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](#), [s813](#), [s818](#), [s821](#), [s853](#),
[s856](#), [s929](#), [s932](#), [s935](#), [s1002](#),
[s1008](#), [t43](#), [t54](#), [t299](#), [t365](#),
[w143](#), [A95](#), [A112](#), [A288](#), [A314](#),
[A326](#), [B359](#), [C219](#), [E88](#), [E97](#),
[H31](#), [K47](#), [N314](#), [N333](#), [N346](#),
[N449](#), [N525](#), [N542](#), [N550](#), [N555](#),
[N583](#), [N638](#), [N975](#), [N1005](#), [I6](#), [I83](#)
`\@latex@info` [d199](#),
[d270](#), [d297](#), [d398](#), [g136](#), [l85](#), [x97](#)
`\@latex@info@no@line` [g136](#), [M576](#)
`\@latex@warning` [g136](#), [g166](#), [l55](#), [o33](#),
[z14](#), [F235](#), [K22](#), [K45](#), [M1996](#),
[N677](#), [N683](#), [N767](#), [N773](#), [B260](#)
`\@latex@warning@no@line`
.. [d177](#), [g136](#), [g167](#), [k17](#), [k74](#),
[k383](#), [z8](#), [z26](#), [z27](#), [A31](#), [H32](#),
[M243](#), [M275](#), [M1827](#), [M2062](#),
[N144](#), [N361](#), [N477](#), [N613](#), [N619](#),
[N642](#), [N726](#), [N733](#), [N800](#), [N877](#)
`\@latexbug` [g225](#), [M333](#), [M1813](#)
`\@latexerr` [g166](#),
[M234](#), [M390](#), [M437](#), [M1874](#), [M1891](#)
`\@lbbitem` [K3](#), [K4](#)
`\@ldots` [u498](#), [u500](#)
`\@leftcolumn` [M121](#),
[M2203](#), [M2224](#), [M2248](#), [M2257](#)
`\@leftmark` [L16](#), [L50](#)
`\@let@token` [d369](#),
[d372](#), [d375](#), [d383](#), [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](#)
`\@linefmt` [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](#),
[M440](#), [M442](#), [M450](#), [M452](#),
[M2161](#), [M2163](#), [M2179](#), [M2180](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- \@makefnmark [I376](#), [I435](#)
- \@makefn_{text} [D333](#), [I418](#)
- \@makeother
 - [a76](#), [a97](#), [a126](#), [d387](#), [d388](#),
 - [o377](#), [o378](#), [o379](#), [o380](#), [o381](#),
 - [o382](#), [o383](#), [o384](#), [o385](#), [o386](#),
 - [o387](#), [A232](#), [A253](#), [A320](#), [A335](#),
 - [A345](#), [N192](#), [N193](#), [N659](#), [N749](#)
- \@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](#), [t131](#), [t232](#)
- \@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
 - [554](#), [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](#), [o135](#),
 - [o136](#), [o160](#), [p7](#), [p445](#), [q372](#),
 - [x78](#), [x114](#), [x778](#), [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](#), [d363](#), [d364](#),
 - [f13](#), [f19](#), [f27](#), [j14](#), [l89](#), [l110](#), [l978](#),
 - [l982](#), [l1013](#), [l1025](#), [l1027](#), [o329](#),
 - [o340](#), [o456](#), [o471](#), [o575](#), [o578](#),
 - [o579](#), [o587](#), [p393](#), [p395](#), [p522](#),
 - [p524](#), [q304](#), [q305](#), [q307](#), [q320](#),
 - [q326](#), [q330](#), [q331](#), [q367](#), [q388](#),
 - [q393](#), [q473](#), [q487](#), [r26](#), [r44](#), [r53](#),
 - [r57](#), [s40](#), [s356](#), [s364](#), [s397](#), [s1013](#),
 - [s1015](#), [w58](#), [w62](#), [E343](#), [E344](#),
 - [N27](#), [N29](#), [N64](#), [N65](#), [N76](#), [N77](#),
 - [N88](#), [N89](#), [N91](#), [N282](#), [N286](#),
 - [N292](#), [N296](#), [N424](#), [N433](#), [N850](#),
 - [N855](#), [N858](#), [N860](#), [N861](#), [N876](#),
 - [N896](#), [N913](#), [N967](#), [N989](#), [N1013](#)
- \@nmbri_{list}false [C33](#), [C46](#), [C91](#)
- \@nmbri_{list}true [C225](#)
- \@nnil [f13](#), [f20](#),
 - [f21](#), [f22](#), [f28](#), [o214](#), [o218](#), [o219](#),
 - [o220](#), [o235](#), [q133](#), [q135](#), [q299](#),
 - [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](#)

File Key: a=lt_{dir}chk.dtx, b=lt_{plain}.dtx, c=lt_{vers}.dtx, d=lt_{defns}.dtx, e=lt_{alloc}.dtx, f=lt_{cntrl}.dtx, g=lt_{error}.dtx, h=lt_{par}.dtx, i=lt_{space}.dtx, j=lt_{logos}.dtx, k=lt_{files}.dtx, l=lt_{outenc}.dtx, m=lt_{counts}.dtx, n=lt_{length}.dtx, o=lt_{fssbas}.dtx, p=lt_{fssaxes}.dtx, q=lt_{fssstrc}.dtx, r=lt_{fssc_{mp}}.dtx, s=lt_{fssdcl}.dtx, t=lt_{fssini}.dtx, u=font_{def}.dtx, v=pre_{load}.dtx, w=lt_{fntcmd}.dtx, x=lt_{textcomp}.dtx, y=lt_{pageno}.dtx, z=lt_{xref}.dtx, A=lt_{miscen}.dtx, B=lt_{math}.dtx, C=lt_{lists}.dtx, D=lt_{boxes}.dtx, E=lt_{tab}.dtx, F=lt_{pictur}.dtx, G=lt_{thm}.dtx, H=lt_{sect}.dtx, I=lt_{float}.dtx, J=lt_{idxglo}.dtx, K=lt_{bibl}.dtx, L=lt_{page}.dtx, M=lt_{output}.dtx, N=lt_{class}.dtx, O=lt_{hyphen}.dtx, P=lt_{luatex}.dtx, Q=lt_{final}.dtx

`\@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, o308, t250, t285, t291, t312, t314, t319
`\@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
`\@nxttabmar` E11, E21, E23, E25, E63, E99, E100, E106, E107
`\@obsoletefile` k382
`\@oddfloat` L11, L14, L15, M124, M610, M669
`\@oddfloat` L11, L14, M123, M610, M669
`\@onefilewithoptions` N388, N392, N398, N416, N420, N426, N439, N488, N826
`\@onelevel@sanitize` .. d389, I42, I111
`\@onlypreamble` d41, d163, d165, d174, d182, k125, k134, k155, k384, k410, l23, l24, l61, l62, l66, l125, l145, l189, l190, l204, o17, o115, o117, o123, o139, o167, o182, o203, o208, o250, o483, 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, s824, s859, s870, s884, s938, s958, s962, s1026, w140, w141, x79, x80, x81, x779, 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, N806, N807, N808, N809, N811
`\@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
`\@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

File Key: a=ltmdirchk.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=ltfssaxes.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

- \@ovhorz [F265](#),
[F266](#), [F279](#), [F309](#), [F310](#), [F322](#)
- \@ovltrue [F248](#), [F293](#)
- \@ovri [D33](#), [F217](#), [F257](#),
[F273](#), [F284](#), [F301](#), [F317](#), [F326](#)
- \@ovro [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](#)
- \@ovvx [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](#),
[N861](#), [N876](#), [N913](#), [N989](#), [N1013](#)
- \@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](#), [N651](#), [N653](#), [N655](#), [N657](#),
[N741](#), [N743](#), [N745](#), [N747](#), [N795](#)
- \@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@videpackage
.. [N148](#), [N150](#), [N168](#), [N170](#), [N181](#)
- \@preamble [E178](#), [E180](#),
[E188](#), [E213](#), [E232](#), [E234](#), [E235](#),
[E239](#), [E254](#), [E272](#), [E273](#), [E310](#)
- \@preamblecmds
... [d41](#), [k64](#), [k119](#), [N1027](#), [N1028](#)
- \@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](#), [d363](#), [g187](#)
- \@qrelax [d112](#), [d363](#)
- \@rc@ifdefinable [d105](#), [d107](#), [d213](#), [I14](#)
- \@reargdef [d97](#)
- \@refundefined [k53](#), [k108](#), [z3](#), [A27](#)
- \@reinserts [M327](#), [M330](#), [M516](#)
- \@remove@tlig [I978](#), [I986](#)
- \@remove@tlig@ [I978](#), [I979](#)
- \@remove@tlig@@ [I979](#), [I982](#)
- \@removeelement [f32](#), [N272](#)
- \@removefromreset
..... [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](#),
[M1414](#), [M1417](#), [M1419](#), [M1493](#),
[M1496](#), [M1499](#), [M1504](#), [M1506](#),
[M1986](#), [M2103](#), [M2108](#), [M2111](#)
- \@reset@ptions [N444](#), [N486](#), [N491](#)
- \@resetactivechars [M575](#), [M598](#), [M656](#)
- \@resethfps ... [M1205](#), [M1374](#), [M2053](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- \@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 [t100](#), [t112](#), [t229](#)
- \@roman [m103](#), [m109](#)
- \@rsbox [D369](#), [D376](#), [D380](#)
- \@rtab [E59](#), [E74](#)
- \@rule [D341](#), [D346](#), [D350](#)
- \@sanitize ... [d387](#), [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](#), [d187](#), [d332](#), [d339](#), [d359](#),
[k257](#), [k296](#), [l131](#), [m156](#), [m161](#),
[x92](#), [x747](#), [x798](#), [x814](#), [z21](#), [L17](#),
[N46](#), [N66](#), [N78](#), [N115](#), [N133](#), [N1018](#)
- \@secpenalty [i17](#), [H36](#), [H50](#)
- \@sect [H54](#), [H55](#)
- \@seqncr [B362](#)
- \@setckpt [k209](#), [k216](#), [A16](#)
- \@setfloattypecounts
... [M1074](#), [M1224](#), [M1395](#),
[M1476](#), [M1571](#), [M1692](#), [M2000](#)
- \@setfontsize [t319](#)
- \@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 [t319](#)
- \@settab [E59](#), [E81](#)
- \@settodim [n17](#)
- \@settopoint [n22](#)
- \@setupverbvisiblepace
... [A274](#), [A292](#), [A295](#), [A307](#)
- \@sffamilyhook [t105](#), [t112](#), [t230](#)
- \@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 [d372](#), [d382](#)
- \@sqrt [B309](#)
- \@ssect [H53](#), [H112](#)
- \@stackcr [F50](#), [F53](#)
- \@star@or@long [d47](#), [d52](#),
[d99](#), [d121](#), [d127](#), [d153](#), [d162](#), [d196](#)
- \@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](#), [x197](#), [x204](#), [x206](#)
- \@tabacol [E166](#), [E234](#)
- \@tabarray [E158](#), [E168](#), [E169](#)
- \@tabclassiv [E168](#), [E306](#)
- \@tabclassz [E167](#), [E258](#)
- \@tabcr [E56](#), [E61](#)
- \@tabfbox [E16](#), [E68](#), [E70](#)
- \@tablab [E60](#), [E114](#)
- \@tabminus [E60](#), [E105](#)
- \@tabplus [E60](#), [E98](#)
- \@tabpush
[E11](#), [E65](#), [E73](#), [E128](#), [E131](#), [E133](#)
- \@tabrj [E60](#), [E112](#)
- \@tabular [E162](#), [E165](#), [E166](#)
- \@tabularcr [E168](#), [E196](#)

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscnp.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

- `\@tempa` z45, z46
`\@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,
N597, N599, N600, N601, N712,
N714, N715, N716, 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, o219,
o224, 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, o220, o225,
o595, o599, 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,
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, I1473, o94, o644, s281,

File Key: a=ltmdirchk.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=ltfssaxes.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=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- s336, s400, s481, s1001, s1007,
A18, A224, A245, K13, M988,
M1024, M1580, M1701, N574, N590
- \@tempstrue a79, b257, k183, k188,
l1476, l1477, o97, o645, o646,
o649, o652, s284, s339, s403,
s484, s964, 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 d191, 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 t110, t112, t231
- \@twoclasseserror N301, N554
- \@twocolumnfalse M99, M147
- \@twocolumntrue M206
- \@twoloadclasserror N484, N549
- \@twosidefalse M100
- \@typein d17, d18, d25, d33
- \@typeset@protect
d77, d218, d225, d227, l26, l32,
l210, l218, t320, A103, A121, Q254
- \@uclclist .. l1442, l1443, l1496, Q461
- \@undefined
.. a68, a69, a108, a109, a110,
a131, a139, a147, a154, a205,
a209, a235, a242, a326, a327,
b65, b81, b105, b106, b121,
b122, b127, b136, b149, b184,
b189, b222, b223, b235, b245,
b280, b448, b491, b537, b538,
c49, c57, d19, d198, d302, d316,
d334, d341, 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,

File Key: a=ltmdirchk.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=ltfssaxes.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

l370, l372, l374, l455, l694, l697,
 m170, o405, o445, o507, o540,
 o604, o611, p577, p578, p580,
 p581, p582, p583, p584, p585,
 p586, p587, p588, p589, p590,
 p591, p592, p593, p594, p595,
 p596, p597, p598, q506, q507,
 r4, r5, r6, r7, r8, r9, r10, r11,
 r12, r13, r14, r15, r16, r17,
 r18, r19, r20, s597, s704, t52,
 t200, t201, t202, t203, t204,
 t205, t206, t207, t208, t209,
 t210, t211, t229, t230, t231,
 t232, t233, t234, t277, t278,
 t279, t280, t281, t282, t293,
 t361, u15, u50, u65, w37, w38,
 w39, w122, x52, x179, x294,
 x536, x537, x567, x568, x569,
 x571, x572, x573, x574, x575,
 x576, x579, x580, x581, x582,
 x585, x586, x587, x588, x590,
 x591, x592, x594, x595, x600,
 x601, x602, x603, x604, x605,
 x606, x607, x608, x609, x610,
 x611, x612, x613, x614, x615,
 x617, x618, x620, x621, x622,
 x623, x624, x625, x626, x627,
 x628, x629, x631, x632, x633,
 x634, x635, x636, x637, x638,
 x639, x641, x642, x643, x644,
 x645, x646, x647, x648, x649,
 x650, x651, x652, x653, x654,
 x655, x656, x657, x658, x659,
 x660, x661, x662, x663, x664,
 x665, x666, x667, x672, x673,
 x675, x676, x677, x678, x679,
 x680, x681, x682, x684, x685,
 x687, x688, x690, x691, x692,
 x693, x695, x696, x697, x699,
 x701, x702, x703, x704, x705,
 x706, x707, x709, x710, x711,
 x712, x713, x714, x715, x716,
 x717, 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, N610, N689,
 N692, N703, N704, N705, N706,
 N707, N708, N709, N780, N783,
 N799, N819, N826, 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
 d195, d230, d236, d241, k139, E240
 \@unknownoptionerror N495, N524, N537
 \@unprocessedoptions
 . . N267, N329, N471, N475, N539
 \@unused d4, g15, g32, g59, k3, N804
 \@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 l150, x16, x1050
 \@vbsphack i193
 \@verb A338, A347, A350
 \@verbatim A214, A260, A294, A303
 \@verbvisiblespacebox
 A279, A280, A283, A308
 \@verreq u451, u452
 \@viipt o687
 \@vipt o686
 \@vipt o685
 \@vline F60, F155
 \@vobeyspaces
 . . A207, A260, A292, A296, A350
 \@vpt o684
 \@vspace i280
 \@vspacer i280
 \@vtryfc M847, M855
 \@vvector F118, F126
 \@warning g166
 \@wckptelt k210, k213
 \@whiledim f7, F36, F83
 \@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,

File Key: a=ltmdirchk.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=ltfssaxes.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

- [D125](#), [F2](#), [F38](#), [F40](#), [F42](#), [F157](#),
[F160](#), [F198](#), [F205](#), [F274](#), [F281](#),
[F318](#), [F324](#), [F365](#), [F366](#), [F404](#)
`\@width` [b394](#),
[d11](#), [i352](#), [l289](#), [l294](#), [q146](#),
[u612](#), [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](#), [o508](#), [o542](#), [o555](#)
`\@wtryfc` [M857](#), [M867](#)
`\@x@protect` [d80](#), [d217](#)
`\@x@sf` [I434](#), [I436](#)
`\@xDeclareMathDelimiter` .. [s769](#), [s825](#)
`\@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` [d385](#)
`\@xdblfloat` [I264](#)
`\@xdim` [F26](#), [F32](#), [F34](#), [F354](#),
[F408](#), [F409](#), [F410](#), [F411](#), [F417](#)
`\@xeqncr` [B341](#)
`\@exnoop` [E214](#), [E224](#)
`\@expast` [E215](#), [E216](#)
`\@xfloat` [I28](#), [I29](#), [I34](#), [I266](#)
`\@xfootnote` [I399](#), [I402](#)
`\@xfootnotemark` [I422](#), [I426](#)
`\@xfootnotenext` [I439](#), [I442](#)
`\@xhline` [E336](#), [E337](#)
`\@xifnch` [d373](#), [d383](#)
`\@xiipt` [o691](#), [u158](#), [u160](#), [u161](#)
`\@xipt` [o690](#), [u157](#)
`\@xivpt` [o692](#), [u159](#), [u161](#)
`\@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` [o689](#), [u156](#), [u159](#), [u160](#)
`\@xsect` [H86](#), [H87](#), [H123](#)
`\@xtabcr` [E56](#), [E57](#)
`\@xtabularcr` [E197](#), [E198](#)
`\@xthm` [G28](#), [G29](#)
`\@xtryfc` [M844](#), [M872](#)
`\@xtypein` [d18](#), [d20](#), [d27](#)
`\@xverbatim` [A209](#), [A260](#)
`\@xviipt` [o693](#), [u160](#), [u162](#)
`\@xxDeclareMathDelimiter` .. [s754](#), [s758](#)
`\@xxpt` [o694](#), [u161](#), [u162](#)
`\@xxvpt` [o695](#), [u162](#)
`\@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](#)
`\[` [o383](#), [t39](#), [t48](#), [B243](#), [B301](#), [B382](#), [Q422](#)
`\` [71](#), [a45](#), [a46](#), [a74](#), [a247](#), [a248](#), [a249](#),
[a250](#), [a253](#), [a260](#), [a261](#), [a262](#),
[a263](#), [a266](#), [a273](#), [a274](#), [a275](#),
[a276](#), [a279](#), [a286](#), [a292](#), [a293](#),
[a297](#), [a299](#), [a300](#), [a304](#), [a309](#),
[a310](#), [a313](#), [a319](#), [b13](#), [d194](#),
[d387](#), [g227](#), [i52](#), [i378](#), [k390](#),
[k405](#), [l557](#), [o371](#), [u245](#), [A179](#),
[A183](#), [A187](#), [A211](#), [B323](#), [B442](#),
[D277](#), [D385](#), [D387](#), [E61](#), [E158](#),
[E168](#), [E182](#), [F50](#), [P259](#), [P411](#), [Q407](#)
`\{` [a3](#), [a7](#), [a74](#), [b2](#), [b13](#), [g22](#), [l308](#), [l559](#),
[o372](#), [u243](#), [A210](#), [B59](#), [B154](#), [Q410](#)
`\}` [a8](#), [a74](#), [b3](#), [b13](#), [g21](#), [l309](#),
[l560](#), [o373](#), [u244](#), [A210](#), [B59](#), [Q411](#)
`\]` [b440](#), [o384](#), [B243](#), [B302](#), [B406](#), [Q423](#)
`\^` ... [a63](#), [a72](#), [a75](#), [a119](#), [a330](#), [b7](#),
[b9](#), [b11](#), [b14](#), [b359](#), [b360](#), [b374](#),
[b375](#), [d5](#), [d388](#), [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](#),

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscnp.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

- 11307, 11308, 11337, 11338, 11359,
 11360, 11361, 11362, o369, o370,
 o375, x184, x201, A48, A57,
 N604, N605, N606, N688, N691,
 N694, N719, N720, N721, N779,
 N782, N785, Q188, Q189, Q190,
 Q191, Q192, Q193, Q194, Q195,
 Q196, Q408, Q414, Q415, Q416,
 Q417, Q451, Q452, Q453, Q454,
 Q455, Q456, Q457, Q458, Q459
 \ a75, b8, b14,
 d388, l314, u248, B223, B224, Q409
 \‘ 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, o385, t418,
 x187, A351, D245, D266, E60, Q424
 \| l558, m135, m146, u567, u568, Q425
 \~ a75, b10, b14, d388, 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, x193,
 x211, A286, A312, A355, Q412

 \sq a74, a91,
 b13, b359, b377, d387, g19, g20,
 g21, g22, g25, i331, o368, o618,
 o654, o679, u246, A207, A208,
 A291, G36, G38, K17, N186, Q402

A
 \A Q185, Q427, Q448
 \a l223, E1, Q176, Q428, Q439
 \AA b365, l240, l425, l523
 \aa b365, l245, l419, l533
 \abovedisplayskip b340, B450
 \abovedisplayskip b339, B443,
 B445, B447, B448, B449, B450
 \accent l73, l393, l420, l476, l757
 \accent@spacefactor l70, l73, l74
 \active a64, a119, a330, b10,
 b11, b374, b375, b377, A207,
 A208, A285, A291, A311, A353,
 B208, B223, M584, N604, N605,
 N606, N688, N691, N694, N719,
 N720, N721, N779, N782, N785
 \active@math@prime B207, B208, M588
 \acute u510
 \add@accent l65, l67
 \add@percent@to@temptokena
 A48, A64, A66, A75
 \add@unicode@accent l999, l1003
 \add_to_callback 588, P699
 \addcontentsline H70, H80, H159, l16
 \addpenalty i220, C124, C170,
 C175, H50, M338, M1153, M1319
 \addto@hook
 o152, o154, o683, s263, s359,
 s363, s380, s504, s510, s518,
 s534, s537, s540, s973, s980, s983
 \addtocontents H164, H171, H174
 \addtocounter l46, m6, m18
 \addtolength l53, m16, B445, B447
 \addtoversion r20, r139
 \addvspace i207, A169, C124,
 C171, C172, C176, C224, H50
 \adjdemerits b319
 \AE l241,
 l398, l524, l762, l1078, l1379, Q480
 \ae l246,
 l401, l534, l766, l1084, l1380, Q480
 \afterassignment b390, b393,
 d231, d237, l212, l220, o299, B186
 \aftergroup
 o91, o313, q156, q222, s114,
 s121, s129, w64, A289, A315,
 D103, M604, M605, M662, M663
 \aleph u302
 \alloc@ b90, b91,
 b92, b93, b94, b95, b96, b97,
 b98, b99, b226, o14, P20, P24, P36
 \allocationnumber
 b37, b57, b69, b71, b143,
 b144, b145, b195, b196, b228,
 b229, b230, b241, b242, b243,
 b260, b266, b272, b273, b286,
 b287, b288, E4, E9, P50, P51,
 P52, P90, P204, Q44, Q45, Q46
 \allowbreak
 b397, d423, d424, d443, d445, B40
 \Alph l46, m106
 \alph l46, m105
 \alpha u262
 \alpha@elt
 s45, s267, s454, s556, s972, s973
 \alpha@list s41, s43, s276, s442, s454,
 s499, s554, s555, s968, s974, s975
 \amalg u373
 \and 438, H14, H27
 \angle u331
 \approx u416
 \arabic l46, m102, G33
 \arccos B13

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\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` u563
`\arrowvert` u561
`\asciispace` A267,
 A269, A272, A275, A276, A305
`\ast` u226, u389
`\asympt` u443
`\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` 438, H8, H24, H32
- B**
- `\b` l233, l389, l472, l753, l1190
`\backslash` u245, u584
`\bar` u514
`\baselineskip`
 b358, b388, b424, q140, q141,
 q142, q144, q145, u504, B158,
 B159, B178, B184, B188, D254,
 D273, E186, F47, F167, M242,
 M273, M622, M637, M681, M696
`\baselinestretch`
 ... o290, 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
`\belowdisplayshortskip` .. b342, B449
`\belowdisplayskip` b341, B448
`\best@size` ... q392, q416, q422, q428
`\beta` u263
`\bezier` 411, F368,
 F369, F423, F424, F439, F441
`\bfdef@ult` t73, t172
`\bfdefault`
 t15, t73, t83, t215, u92, u103, u110
`\bfseries` t13, t14, t77, t213,
 t214, w19, z13, G36, G38, K20
`\bfseries@rm` . t60, t80, t179, t181, t201
`\bfseries@rm@kernel` .. t63, t179, t204
`\bfseries@sf` . t60, t81, t182, t184, t202
`\bfseries@sf@kernel` .. t64, t182, t205
`\bfseries@tt` . t60, t82, t185, t187, t203
`\bfseries@tt@kernel` .. t65, t185, t206
`\bgroup` b372
`\bibcite` K7, K9, K10
`\bibdata` K25, K29
`\bibitem` K3
`\bibliography` 471, K27
`\bibliographystyle` 471, K32
`\bibstyle` K25, K37
`\Big` u617,
 u620, u629, u631, B44, B45, B46
`\big` u618, u630, B41
`\bigbreak` b404, d425, d446
`\bigcap` u339
`\bigcirc` u386
`\bigcup` u340
`\Bigg` u624, u633, B50, B51, B52
`\bigg` u622, u632, B47, B48, B49
`\Biggl` B50
`\biggl` B47
`\Biggm` B51
`\biggm` B48
`\Biggr` B52
`\biggr` B49
`\Bigl` B44
`\bigl` B41
`\Bigm` B45
`\bigm` B42
`\bigodot` u347
`\bigoplus` u346
`\bigotimes` u345
`\Bigr` B46
`\bigr` B43
`\bigskip` b409, i310
`\bigskipamount` .. b408, i312, i313, l367
`\bigsqcup` u350
`\bigtriangledown` u355, u356
`\bigtriangleup` u354, u357
`\biguplus` u338
`\bigvee` u336
`\bigwedge` u337
`\binoppenalty` b310
`\bm@b` D37
`\bm@c` D37
`\bm@l` D37
`\bm@r` D37
`\bm@s` D37
`\bm@t` D37

File Key: a=ltmdirchk.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=ltfssaxes.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

<code>\bmod</code>	B35	<code>\bx@P</code>	M33 , M60
<code>\boldmath</code>	j14 , t312	<code>\bx@PP</code>	M43
<code>\bordermatrix</code>	B172	<code>\bx@Q</code>	M33 , M60
<code>\bot</code>	u315	<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>	u476	<code>\bx@SS</code>	M44
<code>\Box</code>	t371	<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>	u547 , u551 , u552 , u554 , u556	<code>\bx@UU</code>	M44
<code>\bracelu</code>	u549 , u553 , u555	<code>\bx@V</code>	M38
<code>\bracerd</code>	u548 , u553 , u555	<code>\bx@VV</code>	M44
<code>\braceru</code>	u550 , u552 , u556	<code>\bx@W</code>	M39
<code>\bracevert</code>	u602	<code>\bx@WW</code>	M45
<code>\brack</code>	B58	<code>\bx@X</code>	M39
<code>\break</code>	b397 , b402 , d426 , d447 , i89	<code>\bx@XX</code>	M45
<code>\breve</code>	u515	<code>\bx@Y</code>	M39
<code>\brokenpenalty</code>	b315 , o637	<code>\bx@YY</code>	M45
<code>\buildrel</code>	u463 , B149	<code>\bx@Z</code>	M39
<code>\bullet</code>	u375	<code>\bx@ZZ</code>	M25 , M45 , M55
<code>\bx@A</code>	M30 , M57	C		
<code>\bx@AA</code>	M40	<code>\c</code>	I234 , I335 , I337 , I339 , I341 , I343 , I345 , I347 , I349 , I351 , I372 , I374 , I392 , I456 , I475 , I600 , I602 , I627 , I629 , I642 , I668 , I695 , I698 , I699 , I700 , I701 , I702 , I703 , I704 , I705 , I706 , I756 , I1192 , I1206 , I1232 , I1289 , I1290 , I1309 , I1310 , I1313 , I1314 , I1319 , I1320 , I1331 , I1332 , I1339 , I1340 , I1343 , I1344 , x183 , x200
<code>\bx@B</code>	M30 , M57	<code>\c@bottomnumber</code>	...	M2292 , I269 , I274
<code>\bx@BB</code>	M40	<code>\c@dbltopnumber</code>	M2299 , I268 , I283 , I297
<code>\bx@C</code>	M30 , M57	<code>\c@enumi</code>	C227
<code>\bx@CC</code>	M40	<code>\c@enumii</code>	C227 , C227
<code>\bx@D</code>	M30 , M57	<code>\c@enumiv</code>	C227
<code>\bx@DD</code>	M40	<code>\c@equation</code>	B303 , B336 , B462
<code>\bx@E</code>	M30 , M57	<code>\c@errorcontextlines</code>	g163
<code>\bx@EE</code>	M40	<code>\c@footnote</code>	H12 , I373 , I428
<code>\bx@F</code>	M31 , M58	<code>\c@mpfootnote</code>	D303 , I375
<code>\bx@FF</code>	M41	<code>\c@ncel</code>	u455 , u456
<code>\bx@G</code>	M31 , M58	<code>\c@page</code>	y3 , y6 , y7 , M138 , M1818
<code>\bx@GG</code>	M41	<code>\c@secnumdepth</code>	H56 , H71 , H81 , H140
<code>\bx@H</code>	M31 , M58	<code>\c@tocdepth</code>	H140 , H184 , H207
<code>\bx@HH</code>	M41	<code>\c@topnumber</code>	M2288 , I267 , I271
<code>\bx@I</code>	M31 , M58	<code>\c@totalnumber</code>	M2295 , I270 , I276
<code>\bx@II</code>	M41	<code>\cal</code>	t419
<code>\bx@J</code>	M31 , M58	<code>\calculate@math@sizes</code>	o591 , q173
<code>\bx@JJ</code>	M41	<code>\call_callback</code>	589 , P680
<code>\bx@K</code>	M32 , M59	<code>\callback_register</code>	P560
<code>\bx@KK</code>	M42	<code>\callback_descriptions</code>	589 , P808
<code>\bx@L</code>	M32 , M59			
<code>\bx@LL</code>	M42			
<code>\bx@M</code>	M32 , M59			
<code>\bx@MM</code>	M42			
<code>\bx@N</code>	M32 , M59			
<code>\bx@NN</code>	M42			
<code>\bx@O</code>	M33 , M60			
<code>\bx@OO</code>	M43			

File Key: a=ltmdirchk.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=ltfssaxes.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

- \cap u366
- \capitalacute
..... 1833, x180, x197, x600, x852
- \capitalbreve
..... 1840, x181, x198, x601, x859
- \capitalcaron
..... 1839, x182, x199, x602, x858
- \capitalcedilla
..... 1826, x183, x200, x603, x849
- \capitalcircumflex
..... 1834, x184, x201, x604, x853
- \capitaldieresis
..... 1836, x185, x202, x605, x855
- \capitaldotaccent
..... 1842, x186, x203, x606, x861
- \capitalgrave
..... 1832, x187, x204, x607, x851
- \capitalhungarumlaut
..... 1837, x188, x205, x608, x856
- \capitalmacron
..... 1841, x189, x206, x609, x860
- \capitalnewtie
..... 1846, x195, x207, x610, x926, x927
- \capitalogonek
..... 1829, x190, x208, x611, x850
- \capitalring 1838, x191, x209, x612, x857
- \capitaltie
..... 1844, x192, x210, x613, x922, x923
- \capitaltilde
..... 1835, x193, x211, x614, x854
- \caption [I4](#)
- \cases [B153](#), [B154](#), [B164](#), [B166](#)
- \catcodetable [P88](#), [P109](#)
- \catcodetable@atletter [586](#), [P93](#), [P233](#)
- \catcodetable@initex . [586](#), [P93](#), [P230](#)
- \catcodetable@latex .. [586](#), [P93](#), [P232](#)
- \catcodetable@string . [586](#), [P93](#), [P231](#)
- \cdot u388
- \cdotop u496, u502
- \cdots u502
- \cdp@elt [o96](#), [o116](#), [o127](#),
[o128](#), [o149](#), [o152](#), [o154](#), [s201](#),
[s283](#), [s338](#), [s402](#), [s483](#), [Q368](#), [Q369](#)
- \cdp@list [o98](#), [o114](#), [o128](#), [o156](#), [o157](#),
[s219](#), [s285](#), [s340](#), [s404](#), [s485](#), [Q369](#)
- \center [A172](#)
- center (environment) [A172](#)
- \centering
..... [A172](#), [A177](#), [A178](#), [A193](#), [A195](#)
- \centerline [D402](#)
- \cf@encoding [l34](#), [l41](#), [l44](#),
[l51](#), [l154](#), [o256](#), [o266](#), [o276](#), [o297](#)
- \ch@ck [b206](#), [b207](#), [b208](#), [b209](#),
[b227](#), [b237](#), [b238](#), [b239](#), [b240](#),
[b268](#), [b270](#), [b282](#), [b283](#), [b284](#),
[b285](#), [b291](#), [N615](#), [N636](#), [N729](#)
- \changes u115, [N560](#), [N562](#), [N563](#), [N564](#)
- \char [d403](#), [l391](#), [l394](#), [l427](#),
[l430](#), [l441](#), [l448](#), [l474](#), [l478](#), [l483](#),
[l486](#), [l488](#), [l490](#), [l727](#), [l755](#), [l758](#),
[l791](#), [l798](#), [l805](#), [l828](#), [l831](#), [l879](#),
[l988](#), [l1001](#), [l1158](#), [t318](#), [x547](#),
[x554](#), [x556](#), [x670](#), [A267](#), [A356](#),
[B205](#), [F112](#), [F140](#), [F154](#), [F162](#),
[F165](#), [F234](#), [F272](#), [F277](#), [F316](#),
[F320](#), [F335](#), [F336](#), [F338](#), [F349](#)
- \chardef [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](#), [E4](#), [E9](#), [N610](#),
[N728](#), [P20](#), [P24](#), [P36](#), [P45](#),
[P46](#), [P88](#), [P157](#), [P211](#), [Q28](#),
[Q30](#), [Q34](#), [Q53](#), [Q104](#), [Q105](#),
[Q106](#), [Q107](#), [Q108](#), [Q109](#), [Q110](#)
- \chardef@text@cmd [l3](#)
- \charsubdef [Q249](#)
- \charzero [P211](#)
- \check u516
- \check@command [d162](#), [d164](#)
- \check@icl
..... [w9](#), [w44](#), [w49](#), [w55](#), [w63](#), [w70](#), [w72](#)
- \check@icr
..... [w9](#), [w44](#), [w50](#), [w56](#), [w64](#), [w73](#), [w78](#)
- \check@mathfonts ... [j5](#), [l302](#), [l328](#),
[l360](#), [l1196](#), [o319](#), [o321](#), [q204](#), [x584](#)
- \check@nocorr@ [w46](#)
- \check@range [q333](#), [q334](#)
- \check@single [q332](#), [q354](#)
- \CheckCommand [d162](#)
- \CheckEncodingSubset ... [x16](#), [x82](#),
[x130](#), [x131](#), [x132](#), [x176](#), [x178](#),
[x286](#), [x537](#), [x741](#), [x791](#), [x847](#),
[x848](#), [x916](#), [x1033](#), [x1036](#), [x1050](#)
- \chi u282
- \choose [B57](#)
- \circ u385
- \circle [F236](#), [F329](#), [F425](#), [F442](#)
- \citation [K11](#), [K19](#), [K43](#)
- \cite [471](#), [K12](#)
- \cl@ckpt [k210](#), [m35](#)
- \cl@page [y4](#)
- \ClassError [g84](#)
- \ClassInfo [g84](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

`\ClassWarning` [g84](#)
`\ClassWarningNoLine` [g84](#)
`\cleaders` [b438](#), [u542](#), [u545](#)
`\cleardoublepage` [M138](#)
`\clearpage` [k179](#),
[k197](#), [A12](#), [A86](#), [M125](#), [M138](#),
[M143](#), [M200](#), [M407](#), [M410](#),
[M414](#), [M455](#), [M461](#), [M2171](#), [M2188](#)
`\cline` [E343](#)
`\clubpenalty` [b312](#), [k10](#),
[k23](#), [k80](#), [o635](#), [C128](#), [C194](#),
[C196](#), [H100](#), [H106](#), [H130](#), [H135](#)
`\clubsuit` [u325](#)
`\col@number` .. [M95](#), [M148](#), [M208](#), [M220](#)
`\colon` [u497](#)
`\color@begingroup` ... [o614](#), [o674](#),
[B104](#), [B134](#), [D29](#), [D63](#), [D131](#),
[D299](#), [D332](#), [E47](#), [E51](#), [M491](#), [I417](#)
`\color@endbox` .. [D63](#), [M224](#), [M631](#),
[M641](#), [M690](#), [M700](#), [I253](#), [I344](#)
`\color@endgroup` [o619](#), [o680](#), [B104](#),
[B134](#), [D29](#), [D63](#), [D89](#), [D110](#),
[D133](#), [D319](#), [D335](#), [E49](#), [M495](#), [I420](#)
`\color@hbox`
. [D63](#), [M628](#), [M638](#), [M687](#), [M697](#)
`\color@setgroup` [D63](#), [D89](#), [D108](#)
`\color@vbox`
. [D63](#), [M215](#), [I96](#), [I165](#), [I335](#), [I357](#)
`\columnsep` [k25](#), [k82](#), [M81](#), [M202](#)
`\columnseprule` .. [M82](#), [M2226](#), [M2260](#)
`\columnwidth`
. [k22](#), [k25](#), [k26](#), [k28](#), [k79](#), [k82](#),
[k83](#), [k86](#), [D301](#), [D328](#), [M80](#),
[M144](#), [M145](#), [M146](#), [M201](#),
[M202](#), [M203](#), [M204](#), [M205](#),
[M1843](#), [M1845](#), [M2224](#), [M2228](#),
[M2256](#), [M2262](#), [I99](#), [I168](#), [I413](#)
`\conditionally@traceoff` [g239](#)
`\conditionally@traceon` [g239](#)
`\cong` [u451](#)
`\contentsline` [H164](#), [H171](#), [H178](#)
`\coprod` [u335](#)
`\copyright` [I285](#), [I316](#), [t331](#)
`\cos` [B12](#)
`\cosh` [B14](#)
`\cot` [B18](#)
`\coth` [B19](#)
`\count@` [a66](#), [a179](#), [a180](#), [a181](#), [a186](#),
[b41](#), [b191](#), [b192](#), [b197](#), [b199](#),
[b205](#), [b206](#), [b207](#), [b208](#), [b209](#),
[b210](#), [b394](#), [b395](#), [c14](#), [c15](#), [c16](#),
[c17](#), [c18](#), [c20](#), [d144](#), [d148](#), [o648](#),
[o654](#), [o656](#), [q22](#), [q256](#), [q258](#),
[q280](#), [q281](#), [s260](#), [s262](#), [s266](#),
[s585](#), [s586](#), [s587](#), [s633](#), [s634](#),
[s635](#), [s694](#), [s695](#), [s696](#), [s742](#),
[s743](#), [s744](#), [s782](#), [s783](#), [s784](#),
[s790](#), [s791](#), [s792](#), [s836](#), [s837](#),
[s838](#), [s844](#), [s845](#), [s846](#), [s905](#),
[s906](#), [s907](#), [s913](#), [s914](#), [s915](#),
[w115](#), [w118](#), [F406](#), [F407](#), [F408](#),
[F411](#), [F412](#), [F415](#), [F419](#), [N660](#),
[N662](#), [N663](#), [N664](#), [N750](#), [N752](#),
[N753](#), [N754](#), [Q172](#), [Q173](#), [Q180](#),
[Q182](#), [Q435](#), [Q436](#), [Q443](#), [Q445](#)
`\countdef` [a66](#), [b37](#), [b38](#), [b39](#),
[b41](#), [b51](#), [b90](#), [y3](#), [P75](#), [P85](#),
[P174](#), [P182](#), [P190](#), [P198](#), [P212](#), [Q61](#)
`\counterwithin` [I46](#), [m77](#), [m97](#)
`\counterwithin@s` [m77](#), [m78](#), [m98](#)
`\counterwithin@x` [m77](#), [m80](#), [m99](#)
`\counterwithout` [I46](#), [m67](#), [m94](#)
`\counterwithout@s` ... [m67](#), [m68](#), [m95](#)
`\counterwithout@x` ... [m67](#), [m70](#), [m96](#)
`\CountZero` [P212](#)
`\cr` [b368](#), [I497](#),
[I503](#), [I513](#), [I519](#), [x122](#), [x126](#),
[x839](#), [x843](#), [B175](#), [B179](#), [B331](#),
[B361](#), [B459](#), [E180](#), [E191](#), [E198](#),
[E207](#), [E208](#), [E353](#), [F52](#), [F54](#), [F55](#)
`\crrc` [b425](#), [I327](#), [I362](#), [I363](#), [I390](#), [I394](#),
[I397](#), [I473](#), [I477](#), [I481](#), [I483](#), [I486](#),
[I726](#), [I754](#), [I758](#), [I761](#), [I828](#), [I831](#),
[I878](#), [I1195](#), [t333](#), [u331](#), [u332](#),
[u334](#), [u453](#), [u456](#), [u460](#), [u524](#),
[u525](#), [u526](#), [u527](#), [u528](#), [u529](#),
[u531](#), [u532](#), [u533](#), [u534](#), [u535](#),
[u537](#), [x127](#), [x844](#), [B155](#), [B157](#),
[B158](#), [B159](#), [B175](#), [B177](#), [B178](#),
[B179](#), [B197](#), [B198](#), [E159](#), [E160](#), [F52](#)
`\create_callback` [589](#), [P655](#)
`\cs` [u115](#)
`\csc` [B21](#)
`\cup` [u367](#)
`\curr@fontshape`
.. [I180](#), [o88](#), [o334](#), [o342](#), [o346](#),
[o348](#), [o490](#), [o496](#), [o499](#), [o508](#),
[o515](#), [o517](#), [o525](#), [o531](#), [o534](#),
[o542](#), [o549](#), [o551](#), [p417](#), [q92](#),
[q100](#), [q121](#), [q431](#), [q451](#), [q483](#),
[q499](#), [q514](#), [s223](#), [s228](#), [t255](#), [t264](#)
`\curr@math@size`
.... [o323](#), [q210](#), [q216](#), [q221](#), [q238](#)
`\CurrentOption`
... [I1472](#), [I1475](#), [I1480](#), [I1492](#),
[N13](#), [N222](#), [N232](#), [N233](#), [N234](#),
[N239](#), [N246](#), [N247](#), [N248](#), [N251](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

N258, N259, N263, N264, N265, N272, N274, N283, N284, N285, N286, N293, N294, N295, N296, N443, N526, N527, N536, N537			
\CYRA	11444	\CYRGDSC	11450
\cyrA	11444, 11495	\cyrgdsc	11450
\CYRABHCH	11444	\CYRGDSCHCRS	11450
\cyrabhch	11444	\cyrgdschcrs	11450
\CYRABHCHDSC	11444	\CYRGHCRS	11451
\cyrabhchdsc	11444	\cyrghcrs	11451
\CYRABHDZE	11445	\CYRGHK	11451
\cyrabhdze	11444	\cyrghk	11451
\CYRABHHA	11445	\CYRGUP	11451
\cyrabhha	11445	\cyrgup	11451
\CYRAE	11445	\CYRH	11451
\cyrAe	11445	\cyrh	11451
\CYRB	11445	\CYRHDSC	11452
\cyrb	11445	\cyrhdsc	11452
\CYRBYUS	11446	\CYRHHCRS	11452
\cyrbyus	11445	\cyrhhcrs	11452
\CYRC	11446	\CYRHHK	11452
\cyrC	11446	\cyrhhk	11452
\CYRCH	11446	\CYRHRDSN	11453
\cyrch	11446	\cyrhrdsn	11452
\CYRCHLDSC	11446	\CYRI	11453
\cyrchldsc	11446	\cyrI	11453
\CYRCHRDSN	11447	\CYRIE	11453
\cyrchrdsn	11446	\cyrIe	11453
\CYRCHVCRS	11447	\CYRII	11453
\cyrchvcrs	11447	\cyrIi	11453
\CYRD	11447	\CYRISHRT	11453
\cyrD	11447	\cyrishrt	11453
\CYRDELTA	11447	\CYRISHRTDSC	11454
\cyrdelta	11447	\cyrishrtdsc	11454
\CYRDJE	11448	\CYRIZH	11454
\cyrdje	11448	\cyrIzh	11454
\CYRDZE	11448	\CYRJE	11454
\cyrdze	11448	\cyrje	11454
\CYRDZHE	11448	\CYRK	11454
\cyrdzhe	11448	\cyrk	11454
\CYRE	11448	\CYRKBEAK	11455
\cyrE	11448	\cyrkbeak	11455
\CYREPS	11449	\CYRKDSC	11455
\cyrEps	11448	\cyrkdsc	11455
\CYREREV	11449	\CYRKHCRS	11455
\cyrerev	11449	\cyrkhcrs	11455
\CYRERY	11449	\CYRKHK	11456
\cyrery	11449	\cyrkhk	11455
\CYRF	11449	\CYRKVCRS	11456
\cyrf	11449	\cyrkvcrs	11456
\CYRFITA	11450	\CYRL	11456
\cyrfita	11449	\cyrL	11456
\CYRG	11450	\CYRLDSC	11456
\cyrG	11450	\cyrldsc	11456
		\CYRLHK	11457
		\cyrLhk	11456
		\CYRLJE	11457
		\cyrLje	11457

File Key: a=ltDirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspc.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssaxes.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=ltidglo.dtx, K=ltbibl.dtx, L=ltpage.dtx, M=ltoutput.dtx,
N=ltclass.dtx, O=lthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

`\date` 438, [H9](#), [H25](#)
`\day` [a185](#), [N656](#), [N746](#)
`\dblfigrule` [M767](#), [M2325](#)
`\dblfloatpagefraction`
..... [M2302](#), [I287](#), [I301](#)
`\dblfloatsep` [M753](#),
[M765](#), [M1640](#), [M1766](#), [M2309](#)
`\dbltextfloatsep` ... [M222](#), [M230](#),
[M769](#), [M1639](#), [M1765](#), [M2309](#)
`\dbltopfraction` ... [M2301](#), [I284](#), [I298](#)
`\ddag` [I313](#)
`\ddagger` [I313](#),
[m132](#), [m138](#), [m145](#), [m147](#), [u368](#)
`\ddot` [u512](#)
`\ddots` [u507](#)
`\deadcycles` [k203](#), [A39](#), [A86](#), [M299](#)
`\declare@robustcommand` [d196](#)
`\DeclareCurrentRelease`
..... [x724](#), [x727](#), [N981](#)
`\DeclareEmphSequence`
..... [257](#), [t240](#), [t276](#), [t277](#), [t289](#)
`\DeclareEncodingSubset` [x60](#),
[x61](#), [x324](#), [x325](#), [x326](#), [x327](#),
[x328](#), [x329](#), [x330](#), [x331](#), [x332](#),
[x333](#), [x334](#), [x335](#), [x336](#), [x337](#),
[x338](#), [x339](#), [x340](#), [x341](#), [x342](#),
[x343](#), [x344](#), [x345](#), [x346](#), [x347](#),
[x348](#), [x349](#), [x350](#), [x351](#), [x352](#),
[x353](#), [x354](#), [x355](#), [x356](#), [x357](#),
[x358](#), [x359](#), [x360](#), [x361](#), [x362](#),
[x363](#), [x364](#), [x365](#), [x366](#), [x367](#),
[x368](#), [x369](#), [x370](#), [x371](#), [x372](#),
[x373](#), [x374](#), [x375](#), [x376](#), [x377](#),
[x378](#), [x379](#), [x380](#), [x381](#), [x382](#),
[x383](#), [x384](#), [x385](#), [x386](#), [x387](#),
[x388](#), [x389](#), [x390](#), [x391](#), [x392](#),
[x393](#), [x394](#), [x395](#), [x396](#), [x397](#),
[x398](#), [x399](#), [x400](#), [x401](#), [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](#), [x534](#),
[x536](#), [x732](#), [x733](#), [x734](#), [x735](#),
[x774](#), [x781](#), [x782](#), [x783](#), [x784](#),
[x1060](#), [x1061](#), [x1062](#), [x1063](#),
[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](#)
`\DeclareEncodingSubset@aux`
..... [x62](#), [x64](#), [x80](#)
`\DeclareErrorFont` [o448](#), [s232](#), [t380](#), [u49](#)
`\DeclareFixedFont` [o75](#)
`\DeclareFontEncoding`
..... [I377](#), [I460](#), [I710](#), [I732](#), [I738](#), [I824](#),
[I998](#), [o118](#), [u120](#), [u121](#), [u122](#), [u123](#)
`\DeclareFontEncoding@` [o122](#),
..... [o124](#), [o139](#), [Q279](#), [Q299](#), [Q365](#)
`\DeclareFontEncoding@saved`
..... [Q279](#), [Q299](#), [Q381](#)
`\DeclareFontEncodingDefaults` ...
..... [o168](#), [r90](#), [r91](#), [u36](#)
`\DeclareFontFamily` [o93](#), [r85](#), [r86](#)
`\DeclareFontFamilySubstitution` [o408](#)
`\DeclareFontSeriesChangeRule` ...
..... [p5](#), [p6](#), [p8](#), [p9](#), [p10](#),
[p11](#), [p12](#), [p13](#), [p14](#), [p15](#), [p16](#),
[p17](#), [p18](#), [p19](#), [p20](#), [p21](#), [p22](#),
[p23](#), [p24](#), [p25](#), [p26](#), [p27](#), [p28](#),
[p29](#), [p30](#), [p31](#), [p32](#), [p33](#), [p34](#),
[p35](#), [p36](#), [p37](#), [p38](#), [p39](#), [p40](#),
[p41](#), [p42](#), [p43](#), [p44](#), [p45](#), [p46](#),
[p47](#), [p48](#), [p49](#), [p50](#), [p51](#), [p52](#),
[p53](#), [p54](#), [p55](#), [p56](#), [p57](#), [p58](#),
[p59](#), [p60](#), [p61](#), [p62](#), [p63](#), [p64](#),
[p65](#), [p66](#), [p67](#), [p68](#), [p69](#), [p70](#),
[p71](#), [p72](#), [p73](#), [p74](#), [p75](#), [p76](#),
[p77](#), [p78](#), [p79](#), [p80](#), [p81](#), [p82](#),
[p83](#), [p84](#), [p85](#), [p86](#), [p87](#), [p88](#),
[p89](#), [p90](#), [p91](#), [p92](#), [p93](#), [p94](#),
[p95](#), [p96](#), [p97](#), [p98](#), [p99](#), [p100](#),
[p101](#), [p102](#), [p103](#), [p104](#), [p105](#),

File Key: a=lt`dirchk`.dtx, b=lt`plain`.dtx, c=lt`vers`.dtx, d=lt`defns`.dtx,
e=lt`alloc`.dtx, f=lt`cntrl`.dtx, g=lt`error`.dtx, h=lt`par`.dtx, i=lt`space`.dtx,
j=lt`logos`.dtx, k=lt`files`.dtx, l=lt`outenc`.dtx, m=lt`counts`.dtx, n=lt`length`.dtx,
o=lt`fssbas`.dtx, p=lt`fssaxes`.dtx, q=lt`fssstrc`.dtx, r=lt`fsscmap`.dtx, s=lt`fssdcl`.dtx,
t=lt`fssini`.dtx, u=lt`fontdef`.dtx, v=lt`preload`.dtx, w=lt`fntcmd`.dtx, x=lt`textcomp`.dtx,
y=lt`pageno`.dtx, z=lt`xref`.dtx, A=lt`miscen`.dtx, B=lt`math`.dtx, C=lt`lists`.dtx,
D=lt`boxes`.dtx, E=lt`tab`.dtx, F=lt`pictur`.dtx, G=lt`thm`.dtx, H=lt`sect`.dtx,
I=lt`float`.dtx, J=lt`idxglo`.dtx, K=lt`tblbl`.dtx, L=lt`page`.dtx, M=lt`loutput`.dtx,
N=lt`class`.dtx, O=lt`hyphen`.dtx, P=lt`luatex`.dtx, Q=lt`final`.dtx

p106, p107, p108, p109, p110,
 p111, p112, p113, p114, p115,
 p116, p117, p118, p119, p120,
 p121, p122, p123, p124, p125,
 p126, p127, p128, p129, p130,
 p131, p132, p133, p134, p135,
 p136, p137, p138, p139, p140,
 p141, p142, p143, p144, p145,
 p146, p147, p148, p149, p150,
 p151, p152, p153, p154, p155,
 p156, p157, p158, p159, p160,
 p161, p162, p163, p164, p165,
 p166, p167, p168, p169, p170,
 p171, p172, p173, p174, p175,
 p176, p177, p178, p179, p180,
 p181, p182, p183, p184, p185,
 p186, p187, p188, p189, p190,
 p191, p192, p193, p194, p195,
 p196, p197, p198, p199, p200,
 p201, p202, p203, p204, p205,
 p206, p207, p208, p209, p210,
 p211, p212, p213, p214, p215,
 p216, p217, p218, p219, p220,
 p221, p222, p223, p224, p225,
 p226, p227, p228, p229, p230,
 p231, p232, p233, p234, p235,
 p236, p237, p238, p239, p240,
 p241, p242, p243, p244, p245,
 p246, p247, p248, p249, p250,
 p251, p252, p253, p254, p255,
 p256, p257, p258, p259, p260,
 p261, p262, p263, p264, p265,
 p266, p267, p268, p269, p270,
 p271, p272, p273, p274, p275,
 p276, p277, p278, p279, p280,
 p281, p282, p283, p284, p285,
 p286, p287, p288, p289, p290,
 p291, p292, p293, p294, p295,
 p296, p297, p298, p299, p300,
 p301, p302, p303, p304, p305,
 p306, p307, p308, p309, p310,
 p311, p312, p313, p314, p315,
 p316, p317, p318, p319, p320,
 p321, p322, p323, p324, p325,
 p326, p327, p328, p329, p330,
 p331, p332, p333, p334, p335,
 p336, p337, p338, p339, p340,
 p341, p342, p343, p344, p345,
 p346, p347, p348, p349, p350,
 p351, p352, p353, p354, p355,
 p356, p357, p358, p359, p360,
 p361, p362, p363, p364, p365,
 p366, p367, p368, p369, p370,
 p371, p372, p373, p374, p375,
 p376, p377, p378, p379, p380,
 p381, p382, p383, p384, p573, p580
 \DeclareFontSeriesDefault
 [248](#), [t34](#), [t35](#), [t198](#), [t200](#)
 \DeclareFontShape
 [o18](#), [o417](#), [o418](#), [o419](#),
 [o420](#), [o421](#), [o422](#), [o423](#), [o424](#),
 [o425](#), [o426](#), [o427](#), [o428](#), [o429](#),
 [o430](#), [o431](#), [o432](#), [o433](#), [o434](#),
 [o435](#), [o436](#), [o437](#), [r25](#), [r27](#), [r81](#), [r82](#)
 \DeclareFontShape@ [o21](#), [o22](#)
 \DeclareFontShapeChangeRule
 [p444](#), [p458](#), [p459](#),
 [p460](#), [p461](#), [p462](#), [p463](#), [p464](#),
 [p465](#), [p466](#), [p467](#), [p468](#), [p469](#),
 [p470](#), [p471](#), [p472](#), [p473](#), [p474](#),
 [p475](#), [p476](#), [p477](#), [p478](#), [p479](#),
 [p480](#), [p481](#), [p482](#), [p483](#), [p484](#),
 [p485](#), [p486](#), [p487](#), [p488](#), [p489](#),
 [p490](#), [p491](#), [p492](#), [p493](#), [p494](#),
 [p495](#), [p496](#), [p497](#), [p498](#), [p499](#),
 [p500](#), [p501](#), [p502](#), [p503](#), [p504](#),
 [p505](#), [p506](#), [p507](#), [p508](#), [p509](#),
 [p510](#), [p511](#), [p512](#), [p513](#), [p514](#), [p588](#)
 \DeclareFontSubstitution
 [l739](#), [l825](#), [o141](#), [u26](#), [u34](#),
 [u37](#), [u38](#), [u124](#), [u125](#), [u126](#), [u127](#)
 \DeclareMathAccent [s573](#), [u510](#), [u511](#),
 [u512](#), [u513](#), [u514](#), [u515](#), [u516](#),
 [u517](#), [u518](#), [u519](#), [u520](#), [u521](#), [u522](#)
 \DeclareMathAlphabet
 [r119](#), [r123](#), [r125](#), [r132](#),
 [s399](#), [s562](#), [u145](#), [u146](#), [u147](#), [u148](#)
 \DeclareMathAlphabetCharacter . [s748](#)
 \DeclareMathDelimiter
 [s750](#), [u249](#), [u250](#), [u251](#),
 [u252](#), [u253](#), [u254](#), [u257](#), [u259](#),
 [u260](#), [u557](#), [u559](#), [u561](#), [u563](#),
 [u565](#), [u568](#), [u570](#), [u572](#), [u574](#),
 [u576](#), [u578](#), [u580](#), [u582](#), [u584](#),
 [u586](#), [u588](#), [u590](#), [u592](#), [u594](#),
 [u596](#), [u598](#), [u600](#), [u602](#), [u604](#), [u606](#)
 \DeclareMathRadical [s885](#), [u523](#)
 \DeclareMathSizes
 [o205](#), [o211](#), [o233](#), [u151](#),
 [u152](#), [u153](#), [u154](#), [u155](#), [u156](#),
 [u157](#), [u158](#), [u159](#), [u160](#), [u161](#), [u162](#)
 \DeclareMathSizes* [o205](#)
 \DeclareMathSymbol [s686](#), [s749](#),
 [s766](#), [u163](#), [u164](#), [u165](#), [u166](#),
 [u167](#), [u168](#), [u169](#), [u170](#), [u171](#),
 [u172](#), [u173](#), [u174](#), [u175](#), [u176](#),
 [u177](#), [u178](#), [u179](#), [u180](#), [u181](#),
 [u182](#), [u183](#), [u184](#), [u185](#), [u186](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

- 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, u244, u245, u255,
 u256, u258, 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, u318, u319,
 u321, u322, u323, u324, u325,
 u326, u327, u328, u335, u336,
 u337, u338, u339, u340, u341,
 u343, u344, u345, u346, u347,
 u348, u350, u351, u352, u353,
 u354, u355, u358, u359, u360,
 u361, 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, u404, u405, u408, u409,
 u412, u413, u414, u415, u416,
 u417, u418, u419, u420, u421,
 u422, u423, u424, u426, u427,
 u428, u429, u430, u431, u432,
 u435, u436, u437, u439, u440,
 u441, u442, u443, u444, u445,
 u446, u447, u448, u449, u471,
 u473, u495, u496, u497, u547,
 u548, u549, u550, u608, u609, u610
 \DeclareMathVersion s245, t2, t3
 \DeclareOldFontCommand . . w125, w141
 \DeclareOption 553, l1471, q29, q37,
 q45, q53, q56, q60, x732, x733,
 x734, x735, x736, x737, x738,
 x739, x740, x781, x782, x783,
 x784, x785, x787, x788, N205, N547
 \DeclareOption* 553, N205
 \DeclarePreloadSizes
 o185, r95, r96, v19, v21,
 v22, v23, v25, v26, v27, v28,
 v29, v30, v34, v38, v43, v45,
 v49, v50, v53, v54, v57, v58, v64
 \DeclareRelease x723, x726, N888
 \DeclareRobustCommand
 d196, d393, d394,
 d401, 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, l310, l311, l312, l313,
 l314, l316, l318, l320, m155,
 n19, n20, n21, o251, o279, o280,
 o281, o282, o287, o289, o307,
 o612, o621, p386, p387, p446,
 p450, p454, p515, p516, p575,
 p576, q113, t4, t7, t10, t13,
 t16, t19, t22, t25, t28, t77, t87,
 t97, t102, t107, t213, t216, t219,
 t222, t225, t248, t249, t284,
 t290, t312, t314, t318, t331,
 t339, t353, u329, u330, u331,
 u342, u349, u406, u407, u438,
 u450, u454, u457, u462, u464,
 u466, u469, u472, u474, u475,
 u477, u479, u481, u483, u485,
 u487, u489, u491, u493, u499,
 u501, u503, u506, u524, u527,
 u530, u534, u538, u541, u544,
 u551, u554, u611, u612, u613,
 u618, u620, u622, u624, w3,
 w126, x4, x11, x45, x1045, 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,

File Key: a=ltmdirchk.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=ltfssaxes.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

H201, K12, L22, L30, L53, L55,
 L57, L64, M87, N514, N515,
 N1011, Q461, Q471, I378, I386
 \DeclareSizeFunction
 q371, q444, q445,
 q456, q457, q461, q462, q468,
 q469, q497, q511, q512, q519, q520
 \DeclareSymbolFont
 r136, s280, u135, u136, u137, u138
 \DeclareSymbolFontAlphabet
 s959, u142, u143, u144
 \DeclareSymbolFontAlphabet@ s960, s963
 \DeclareTextAccent l64, I378,
 I379, I380, I381, I382, I383, I384,
 I385, I386, I387, I388, I461, I462,
 I463, I464, I465, I466, I467, I468,
 I469, I470, I471, I735, I740, I741,
 I742, I743, I744, I745, I746, I747,
 I748, I749, I750, I832, I833, I834,
 I835, I836, I837, I838, I839, I840,
 I841, I842, I843, I844, I845, I846
 \DeclareTextAccentDefault
 l185, I226, I227, I228,
 I229, I230, I231, I232, I233, I234,
 I235, I236, I237, I238, I239, I279,
 I282, x597, x598, x849, x850,
 x851, x852, x853, x854, x855,
 x856, x857, x858, x859, x860, x861
 \DeclareTextCommand
 l3, I58, I65, I83, I389,
 I392, I395, I411, I412, I419, I421,
 I423, I425, I431, I433, I435, I442,
 I472, I475, I479, I482, I484, I487,
 I489, I491, I507, I724, I751, I753,
 I756, I759, I792, I799, I826, I829,
 I876, I1003, I1029, I1031, I1033,
 I1157, I1194, x303, x304, x305,
 x306, x307, x308, x309, x310,
 x311, x312, x541, x548, x555, x669
 \DeclareTextCommandDefault
 l57, I186, I188, I283, I286, I287,
 I288, I290, I292, I296, I300, I301,
 I303, I304, I305, I306, I326, I355,
 x175, x177, x180, x181, x182,
 x183, x184, x185, x186, x187,
 x188, x189, x190, x191, x192,
 x193, x194, x195, x214, x215,
 x216, x217, 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, x264, x265, x266,
 x267, x268, x269, x270, x271,
 x272, x273, x274, x275, x276,
 x277, x278, x279, x280, x281,
 x282, x283, x284, x285, x287,
 x288, x289, x290, x291, x292,
 x293, x570, x577, x578, x583,
 x589, x915, x917, x918, x920,
 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, x1035
 \DeclareTextComposite
 l76, I449, I450, I581,
 I582, I583, I584, I585, I586, I587,
 I588, I589, I590, I591, I592, I593,
 I594, I595, I596, I597, I598, I599,
 I600, I601, I602, I603, I604, I605,
 I606, I607, I608, I609, I610, I611,
 I612, I613, I614, I615, I616, I617,
 I618, I619, I620, I621, I622, I623,
 I624, I625, I626, I627, I628, I629,
 I630, I631, I632, I633, I634, I635,
 I636, I637, I638, I639, I640, I641,
 I642, I643, I644, I645, I646, I647,
 I648, I649, I650, I651, I652, I653,
 I654, I655, I656, I657, I658, I659,
 I660, I661, I662, I663, I664, I665,
 I666, I667, I668, I669, I670, I671,
 I672, I673, I674, I675, I676, I677,
 I678, I679, I680, I681, I682, I683,
 I684, I685, I686, I687, I688, I689,
 I690, I691, I806, I807, I808, I809,
 I810, I811, I812, I813, I814, I815,
 I816, I817, I818, I819, I820, I821
 \DeclareTextCompositeCommand ...
 l76, I428, I451,
 I452, I453, I454, I456, I692, I693,
 I695, I698, I699, I700, I701, I702,
 I703, I704, I705, I706, I789, I1009
 \DeclareTextFontCommand
 w1, w15, w16, w17,
 w18, w19, w20, w21, w22, w23,
 w24, w29, w30, w31, w42, w140

File Key: a=ltmdirchk.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=ltfssaxes.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

<code>\DeclareTextSymbol</code>	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, x295, x296, x297, x298, x299, x300, x301, x302, x313, x314, x315, x316, x317, x318, x319, x320, x321, x322, x515, x516, x517, x518, x519, x520, x521, x522
<code>\DeclareTextSymbolDefault</code>	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, x140, x141, x142, x143, 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, x503, x504, x505, x506, x507, x508, x509, x510, x523, x524, x525, x526, x527, x528, x529, x530, x539, x540, x558, x559, x560, x561, x562, x563, x564, x566, x862, x863, x864, x865, 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
<code>\DeclareUnicodeAccent</code>	l1002, l1179, l1180, l1181, l1182, l1183, l1184, l1185, l1186, l1187, l1188, l1189, l1190, l1191, l1192, l1193
<code>\DeclareUnicodeCharacter</code>	Q280
<code>\DeclareUnicodeComposite</code>	

File Key: a=ltmdirchk.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=ltfssaxes.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

- [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](#)
- [\default@family](#) [o129](#), [o161](#),
[o458](#), [o472](#), [o475](#), [o500](#), [o535](#), [Q370](#)
- [\default@M](#) [o136](#), [o176](#), [o179](#), [o183](#), [Q377](#)
- [\default@mextra](#) [r10](#), [r89](#)
- [\default@series](#) [o129](#), [o162](#),
[o459](#), [o473](#), [o476](#), [o497](#), [o532](#), [Q370](#)
- [\default@shape](#) [o130](#), [o163](#),
[o460](#), [o474](#), [o477](#), [o495](#), [o530](#), [Q371](#)
- [\default@T](#) [o170](#), [o173](#), [o183](#), [o272](#)
- [\defaultthyphenchar](#) [b322](#), [d404](#)
- [\defaultscritratio](#) [o595](#), [o602](#)
- [\defaultscriptscritratio](#) [o596](#), [o602](#)
- [\defaultskewchar](#) [b323](#)
- [\define@mathalphabet](#) [r18](#), [r131](#)
- [\define@mathgroup](#) [r19](#), [r135](#)
- [\define@newfont](#) [o326](#), [o335](#)
- [\deg](#) [B34](#)
- [\delcode](#) [s883](#)
- [\delimiter](#) [s796](#), [s866](#), [s877](#)
- [\delimiterfactor](#) [b324](#)
- [\delimitershortfall](#) [b334](#)
- [\Delta](#) [u292](#)
- [\delta](#) [u265](#)
- [\depth](#) [D32](#), [D35](#)
- [\det](#) [B30](#)
- [\detokenize](#) [l980](#), [l1000](#), [p396](#),
[t262](#), [x65](#), [A51](#), [N626](#), [N627](#), [N631](#)
- [\development@branch@name](#) ... [c11](#),
[c36](#), [c49](#), [c50](#), [c51](#), [c57](#), [c58](#), [c59](#)
- [\DH](#) [l525](#), [l1079](#), [Q481](#)
- [\dh](#) [l535](#), [l1085](#), [Q481](#)
- [\Diamond](#) [t372](#)
- [\diamond](#) [u374](#)
- [\diamondsuit](#) [u326](#)
- [\dim](#) [B28](#)
- [\dimen@](#) [b41](#), [b391](#), [b392](#), [b428](#),
[b429](#), [b431](#), [b433](#), [g28](#), [g29](#),
[i295](#), [i300](#), [l426](#), [l427](#), [l429](#), [l430](#),
[l790](#), [l791](#), [o214](#), [o216](#), [o222](#),
[o235](#), [o238](#), [o242](#), [o594](#), [o595](#),
[o596](#), [o600](#), [q405](#), [q406](#), [q407](#),
[q408](#), [q412](#), [x123](#), [x125](#), [x840](#),
[x842](#), [B72](#), [B73](#), [B186](#), [B187](#),
[B188](#), [B189](#), [D394](#), [D397](#), [E164](#),
[E165](#), [M508](#), [M510](#), [M531](#), [M533](#)
- [\dimen@i](#) [b41](#)
- [\dimen@ii](#) [b41](#), [o218](#), [o223](#)
- [\dimendef](#) [b42](#), [b43](#), [b44](#), [b52](#), [b91](#), [P213](#)
- [\dimenzero](#) [P213](#)
- [\dimexpr](#) [l500](#), [l516](#), [l1161](#), [l1164](#)
- [\directlua](#) [a9](#), [a12](#),
[a17](#), [a20](#), [a25](#), [b65](#), [b81](#), [b105](#),
[b245](#), [d19](#), [d341](#), [N610](#), [P2](#), [P12](#),
[P27](#), [P204](#), [P218](#), [P243](#), [P248](#), [P252](#)
- [\disable_callback](#) [589](#), [P800](#)
- [\discretionary](#) [d402](#), [d414](#), [B205](#)
- [\disply@y](#) [B191](#), [B195](#), [B196](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

<code>\displaylines</code>	B190	<code>\dump</code>	Q548
<code>\displaymath</code>	B301	E	
<code>displaymath</code> (environment)	B299	<code>\E</code>	N666, N669, N696, N756, N759, N787
<code>\displaystyle</code>	u526, u529, u533, u535, B62, B197, B325, B328, B365, B390, B402, B430, B454, B457	<code>\e@alloc</code> ..	b51, b52, b53, b55, b56, b63, b64, b66, b68, b79, b82, b84, b138, P13, P47, P78, P88, P177, P185, P193, P201, Q12, Q33
<code>\displaywidowpenalty</code>	b314	<code>\e@alloc@attribute@count</code> P66, P74, P75, P79, P81, P224
<code>\displaywidth</code>	B197, B324, B377, B433	<code>\e@alloc@bytecode@count</code>	P70, P189, P190, P194, P196, P240
<code>\div</code>	u377	<code>\e@alloc@ccodetable@count</code> P67, P84, P85, P89, P92, P228
<code>\DJ</code>	l526, l1089, Q481	<code>\e@alloc@chardef</code> b60, b102, b210, b211, P46, P177, P185, P193, P201, Q12
<code>\dj</code>	l536, l1090, Q481	<code>\e@alloc@intercharclass@top</code>	Q21
<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, N579, N659, N749, N1026, I65, I134	<code>\e@alloc@luachunk@count</code>	P71, P197, P198, P202, P206, P242
<code>\do@add@percent@to@temptokena</code> A52, A58, A76	<code>\e@alloc@luafunction@count</code> P68, P173, P174, P178, P180, P234, P236
<code>\do@emfont@update</code>	t256, t260, t280	<code>\e@alloc@top</code>	b55, b63, b102, b188, b248, P45, P79, P178, P186, P194, P202, Q12
<code>\do@noligs</code>	A352, A357	<code>\e@alloc@whatsit@count</code>	P69, P181, P182, P186, P188, P238
<code>\do@subst@correction</code> ..	o84, q436, q491	<code>\e@ch@ck</code>	b142, b152, P49, P53
<code>\DocInput</code>	q8, u5, v5, O4	<code>\e@insert@top</code> ..	b246, b248, b265, b280
<code>\document</code>	88, k11, K40	<code>\e@mathgroup@top</code> ..	b79, b124, s56, s145
<code>\document@default@language</code> k49, k50, M597, Q217	<code>\egroup</code>	b372
<code>\document@select@group</code> s137, s236	<code>\eject</code>	b402
<code>\documentclass</code> q2, u2, v2, N300, N307, N334, N337, N460, N556, O2, P14	<code>\ell</code>	u305
<code>\documentstyle</code>	N305, N556	<code>\em</code>	t249, t276, w42
<code>\dont@add@percent@to@temptokena</code> A51, A53, A77	<code>\em@currfont</code>	t255, t266
<code>\dorestore@version</code>	s114, s119	<code>\em@force</code>	t264, t269, t272, t282
<code>\dospecials</code> ..	a74, a126, b13, A232, A253, A335, A345, N659, N749	<code>\emergencystretch</code>	L59, L65
<code>\dot</code>	u519	<code>\emfontdeclare@clist</code>	t245, t247, t251, t256, t261, t267, t278
<code>\doteq</code>	u463	<code>\emforce</code>	257, t245, t272, t281
<code>\dotfill</code>	b435, d427, d448	<code>\eminnershape</code>	t253, t259, t276
<code>\dots</code>	l320, l322	<code>\emph</code>	w42
<code>\doublehyphendemerits</code>	b317	<code>\empty</code>	b370
<code>\doublerulesep</code>	E287, E314, E338	<code>\empty@sfcnt</code>	q444, q445, q446, q460, q465, q517
<code>\Downarrow</code>	u580	<code>\emptyset</code>	u312
<code>\downarrow</code>	u574	<code>\emreset</code>	257, t245, t248, t279
<code>\downbracefill</code>	u532, u551	<code>\emrest</code>	t248
<code>\ds@</code>	N227, N497	<code>\enc@update</code> ..	o257, o259, o275, o278, q129
<code>\dt@pfalse</code>	B192	<code>\encodingdefault</code> 1973, l1472, l1505, l1506, s237, t340, t354, u50, P247, P262, P270
<code>\dt@ptrue</code>	B191		

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\end` . a69, d8, d363, g200, q9, u6, v6,
A92, [A118](#), A211, A212, B411,
B420, C112, H15, H17, N674,
N678, N684, N764, N768, N774, 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, d274, d300,
d303, d319, d352, d361, d412,
d416, d441, d462, 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, o52, o73, o231,
o248, o401, o406, o441, o446,
o463, o481, o520, o553, o669,
o681, p571, p601, q503, q509,
r21, r143, s77, s105, s168, s198,
s622, s664, s677, s684, s872,
s880, t196, t238, t275, t287,
t294, t349, t363, u80, u90, u105,
u111, u627, u634, w33, w40,
x41, x55, x532, x719, 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, N699,
N789, 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 [551](#)
flushleft [A202](#)
flushright [A204](#)
itemize [C242](#)
lrbox [376](#)
math [B299](#)
minipage [377](#)
sloppypar [L62](#)
thebibliography [471](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- `verbatim*` [A294](#)
`\epsilon` [u266](#)
`\eqnarray` [B316](#), [B363](#)
`eqnarray` (environment) ... [B311](#), [B436](#)
`eqnarray*` (environment) [B362](#)
`\eqno` [B305](#)
`\equation` [B304](#)
`equation` (environment) ... [B303](#), [B424](#)
`\equiv` [u442](#)
`\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](#), [o492](#), [o527](#), [q379](#),
[q479](#), [r65](#), [O16](#), [P61](#), [Q49](#), [Q239](#)
`\error@fontshape` [o453](#),
[o468](#), [o493](#), [o528](#), [q107](#), [q481](#), [s222](#)
`\errorcontextlines` [b327](#),
[b454](#), [b470](#), [b485](#), [b498](#), [b515](#), [g163](#)
`\errorstopmode` [b443](#), [Q547](#)
`\escapechar` [d101](#), [d144](#), [d148](#),
[d156](#), [k228](#), [o338](#), [o562](#), [q183](#),
[s58](#), [s86](#), [s147](#), [s177](#), [s221](#), [P203](#)
`\et@xmaxfam` [P20](#), [P24](#), [P28](#), [P36](#)
`\et@xmaxregs` [P27](#),
[P29](#), [P30](#), [P31](#), [P32](#), [P33](#), [P34](#), [P35](#)
`\eta` [u268](#)
`\etatcatcode` [P852](#)
`\eTeXversion` [a57](#)
`\evensidemargin` ... [M73](#), [M613](#), [M672](#)
`\every@math@size` ... [o78](#), [q189](#), [q201](#)
`\everycr` [b423](#), [B192](#), [B195](#), [B324](#), [B451](#)
`\everydisplay` [o316](#), [o317](#), [o322](#)
`\everyjob` [c37](#), [c42](#),
[c47](#), [s241](#), [P207](#), [P208](#), [P249](#),
[P250](#), [Q312](#), [Q506](#), [Q507](#), [Q509](#)
`\everymath` [o315](#), [o317](#), [o320](#)
`\everypar` [69](#), [k41](#), [k99](#),
[o615](#), [o628](#), [o675](#), [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](#), [x752](#), [x789](#), [N276](#)
`\exhyphenpenalty` [b309](#), [b396](#)
`\exists` [u318](#)
`\exp` [B31](#)
`\expand@font@defaults` [t69](#),
[t79](#), [t89](#), [t126](#), [t188](#), [t210](#), [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`
..... [o568](#), [o574](#), [s151](#), [s182](#)
`\extract@default@composite`
..... [I1012](#), [I1019](#)
`\extract@default@composite@a` ...
..... [I1021](#), [I1025](#)
`\extract@default@composite@b` ...
..... [I1023](#), [I1027](#)
`\extract@font` [o349](#), [q81](#)
`\extract@fontinfo` [q312](#), [q319](#)
`\extract@rangefontinfo`
..... [q329](#), [q336](#), [q355](#), [q388](#)
`\extract@sizefn` [q304](#), [q326](#)
`\extrafloats` [b152](#), [b189](#), [b263](#)
- F**
- `\f@baselineskip`
..... [I1162](#), [o288](#), [o295](#), [o479](#), [q119](#),
[q136](#), [q140](#), [q155](#), [q169](#), [q180](#), [q194](#)
`\f@depth` [M345](#), [I291](#)
`\f@encoding` [I178](#), [o251](#), [o270](#),
[o273](#), [o274](#), [o276](#), [o297](#), [o329](#),
[o334](#), [o353](#), [o355](#), [o357](#), [o362](#),
[o364](#), [o395](#), [o457](#), [o489](#), [o524](#),
[p402](#), [p406](#), [p530](#), [p534](#), [q91](#),
[q261](#), [q471](#), [s207](#), [P247](#), [P262](#), [P270](#)
`\f@family` [o279](#), [o283](#), [o291](#),
[o330](#), [o334](#), [o353](#), [o355](#), [o357](#),
[o362](#), [o364](#), [o396](#), [o475](#), [o500](#),
[o535](#), [p402](#), [p406](#), [p530](#), [p534](#),
[q91](#), [s207](#), [t80](#), [t81](#), [t82](#), [t90](#), [t91](#),
[t92](#), [t129](#), [t155](#), [t156](#), [t341](#), [x23](#),
[x27](#), [x29](#), [x31](#), [x84](#), [x87](#), [x101](#),
[x105](#), [x107](#), [x109](#), [x114](#), [x121](#),
[x806](#), [x809](#), [x823](#), [x832](#), [x838](#), [x1053](#)
`\f@linespread` ... [o291](#), [q118](#), [q137](#),
[q138](#), [q141](#), [q149](#), [q152](#), [q163](#), [q166](#)
`\f@series` [j14](#),
..... [o279](#), [o284](#), [o292](#), [o331](#), [o334](#),
[o476](#), [o497](#), [o532](#), [p386](#), [p391](#),
[p425](#), [p426](#), [p530](#), [p534](#), [p575](#),
[t122](#), [t138](#), [t140](#), [t144](#), [t145](#),
[t164](#), [t167](#), [t170](#), [t172](#), [t342](#), [x7](#), [x48](#)
`\f@shape` [o279](#), [o285](#),
[o293](#), [o332](#), [o334](#), [o477](#), [o495](#),
[o530](#), [p402](#), [p406](#), [p516](#), [p520](#),
[p527](#), [p532](#), [p535](#), [p538](#), [p576](#), [t343](#)
`\f@size` [I180](#), [I1161](#), [o88](#),
..... [o288](#), [o294](#), [o333](#), [o478](#), [o517](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

- o551, o593, o594, o597, o598,
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, t255, t264, x123, x840
- \f@user@size .. q427, q432, q476, q489
- \fam b98, o15, P20, P24, P36
- \famdef@ult t75, t189, t190, t191
- \familydefault
. i1506, s238, t75, t341, t355, u113
- \fbox 376, D128, D141, D148
- \fboxrule D126, D162, D165,
D171, D173, D180, D181, Q84
- \fboxsep D126, D132,
D161, D166, D176, D178, Q83
- \filbreak b400
- \file@ntents
... N569, N572, N575, N586,
N607, N701, N723, N811, N1026
- \file@ntents@checkdir
..... N592, N594, N608, N708
- \file@ntents@force N588, N704
- \file@ntents@noheader .. N590, N706
- \file@ntents@nosearch .. N591, N707
- \file@ntents@opt
..... N572, N575, N577, N703
- \file@ntents@overwrite . N589, N705
- \file@ntents@where
..... N593, N595, N620, N709
- \filecontents N560
- filecontents (environment) 551
- \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,

File Key: a=ltmdirchk.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=ltfssaxes.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

- 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](#) [u322](#)
[\float@count](#) .. [b51](#), [b52](#), [b53](#), [b62](#),
[b188](#), [b205](#), [b210](#), [b212](#), [b213](#), [b222](#)
[\floatingpenalty](#) [I412](#)
[\floatpagefraction](#) [M2298](#), [I278](#)
[\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](#)
[\fmtversion](#) [c1](#),
[c19](#), [c38](#), [c40](#), [c43](#), [c45](#), [c48](#), [c56](#),
[c84](#), [g2](#), [I1474](#), [t194](#), [E1](#), [F1](#),
[M4](#), [N360](#), [N363](#), [P263](#), [Q490](#), [Q516](#)
[\fmtversion@topatch](#) [Q488](#),
[Q490](#), [Q502](#), [Q503](#), [Q515](#), [Q523](#)
[\fnsymbol](#) [I46](#), [m107](#)
[\font](#) [b428](#), [b433](#),
[d403](#), [d406](#), [I297](#), [I298](#), [I299](#),
[I436](#), [I443](#), [I793](#), [I800](#), [I985](#), [I986](#),
[I1010](#), [I1158](#), [o81](#), [o87](#), [o89](#), [q84](#),
[t252](#), [t285](#), [t291](#), [t317](#), [v8](#), [v9](#),
[v10](#), [w85](#), [x6](#), [x47](#), [x542](#), [x549](#), [A255](#)
[\font@info](#) [q99](#), [q319](#), [q388](#), [q393](#)
[\font@name](#) [I179](#),
[I182](#), [o86](#), [o194](#), [o196](#), [o325](#),
[o340](#), [o516](#), [o550](#), [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](#),
[I297](#), [I298](#), [I299](#), [I436](#), [I443](#), [I793](#),
[I800](#), [t252](#), [t285](#), [t291](#), [w85](#), [x6](#),
[x47](#), [x542](#), [x549](#), [F38](#), [F40](#), [F365](#)
[\fontencoding](#)
..... [I1505](#), [o251](#), [o282](#), [s237](#),
[t340](#), [u16](#), [u24](#), [u67](#), [u74](#), [u83](#),
[u85](#), [x36](#), [P246](#), [P247](#), [P269](#), [P270](#)
[\fontfamily](#) [o279](#),
[s238](#), [t6](#), [t9](#), [t12](#), [t123](#), [t134](#),
[t221](#), [t224](#), [t227](#), [t383](#), [u58](#),
[u69](#), [u76](#), [u87](#), [x28](#), [x30](#), [x32](#),
[x34](#), [x106](#), [x108](#), [x110](#), [x112](#), [x825](#)
[\fontname](#) [I986](#), [o89](#)
[\fontseries](#) [o279](#), [p387](#), [p575](#),
[s239](#), [t15](#), [t18](#), [t80](#), [t81](#), [t82](#), [t83](#),
[t90](#), [t91](#), [t92](#), [t93](#), [t215](#), [t218](#), [t384](#)
[\fontseriesforce](#) [p386](#), [p577](#)
[\fontshape](#) [I446](#),
[I803](#), [o279](#), [p448](#), [p452](#), [p456](#),
[p515](#), [p546](#), [p550](#), [p553](#), [p556](#),
[p559](#), [p562](#), [p565](#), [p568](#), [p576](#),
[s240](#), [t21](#), [t24](#), [t27](#), [t30](#), [t385](#), [x552](#)
[\fontshapeforce](#) [p516](#), [p578](#)
[\fontsize](#) [j6](#), [I302](#), [I328](#), [I360](#),
[I1160](#), [I1196](#), [o79](#), [o289](#), [t323](#),
[t386](#), [x125](#), [x584](#), [x842](#), [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](#) [u317](#)
[\fps@dbl](#) [I34](#)
[\frac](#) [B308](#)
[\frame](#) [D112](#), [D188](#)
[\framebox](#) [376](#), [D135](#)
[\frenchspacing](#) [b353](#), [d428](#),
[d449](#), [k44](#), [k102](#), [A260](#), [A296](#), [A350](#)
[\frown](#) [u445](#)
[\frozen@everydisplay](#) [o315](#), [o321](#)
[\frozen@everymath](#) [o315](#), [o319](#)
[\fussy](#) [L64](#)
[\futurelet](#) [d369](#),
[d383](#), [i320](#), [i328](#), [w83](#), [B210](#), [E335](#)
- G**
- [\g@addto@macro](#) [N504](#), [N510](#), [N514](#), [N515](#)
[\G@refundefinedfalse](#) [z5](#)
[\G@refundefinedtrue](#) [z3](#), [z12](#), [K21](#), [K44](#)
[\Gamma](#) [u291](#)
[\gamma](#) [u264](#)
[\gcd](#) [B33](#)
[\ge](#) [u411](#), [u413](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- \gen@sfcnt q456, q457, q458
 \genb@sfcnt q461, q462, q463
 \genb@x q464, q466
 \genb@y q466
 \GenericError [g18](#), g85, g111, g137, q62
 \GenericInfo
 . c85, c88, c93, [g4](#), g104, g130,
 [g155](#), q31, q34, q39, q75, N952
 \GenericWarning [g11](#),
 [g94](#), g120, g146, q42, q47, q50, q78
 \geq u409, u411
 \get@cdp s356, s364, [s397](#)
 \get@external@font q83, [q96](#), q490
 \getanddefine@fonts .. o563, o581,
 [q274](#), s59, s87, s132, s148, s178,
 s263, s327, s361, s363, s380,
 s503, s504, s536, s537, s979, s980
 \GetFileInfo u3
 \gets u433, u435
 \gg u427
 \glb@currsz k39, k97,
 o312, [q171](#), q206, q210, q216, q239
 \glb@settings . o313, [q171](#), q218, q249
 \globaldefs
 . . o564, q185, s60, s89, s149, s180
 \glossary 469, H176,
 J23, [J35](#), L24, L32, M621, M680
 \glossaryentry J32
 \goodbreak [b400](#), d429, d450
 \grave u511
 \group@elt s35,
 s261, s298, s299, [s320](#), s324, s1011
 \group@list s265,
 s305, [s318](#), s323, s324, s353,
 s579, s627, s688, s773, s776,
 s827, s830, s896, s899, s966, s1017
 \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, x188, x205
 \h@false B81
 \h@true B82, B83
 \halign . b423, B127, B197, B324, B451
 \hangindent H139
 \hat u517
 \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, o618, o625, o660, o679
 \hbar u329
 \headheight M75, M626, M685
 \headsep M76, M635, M694
 \heartsuit u327
 \height l1164, D31, D34
 \hexnumber@ s600,
 s608, s643, s651, s672, s682,
 s708, s716, s724, s733, s736,
 s745, s746, s785, s793, s839,
 s847, s866, s867, s877, s878,
 s883, s909, s917, s922, s924, t327
 \hfuzz . b328, o626, 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, x541, x548
 \hmode@start@before@group
 l68, l151, l153, l159, [l184](#)
 \hom B29
 \hookleftarrow u474
 \hookrightarrow u472
 \hphantom [B75](#)
 \hrule b391,
 b435, i296, i304, l289, l294,
 u334, u612, D118, D123, D171,
 D181, E335, E352, F281, F324, l371
 \hrulefill [b435](#), d430, d451
 \hspace [i350](#)
 \Hwithstroke l491, l1155
 \hwithstroke l507, l1156
 \hyphenation [l205](#)
 \hyphenchar d396, d403, d406, d413, A255
 \hyphenpenalty b308, o632, o664

File Key: a=ltmdirchk.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=ltfssaxes.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

I

- \I [b359](#), [N692](#), [N783](#), [N803](#), [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](#),
[u331](#), [u453](#), [u524](#), [u527](#), [u531](#),
[u534](#), [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](#), [N618](#), [N637](#)
- \if@firststamp [E227](#)
- \if@firstcolumn
... [M95](#), [M246](#), [M279](#), [M396](#),
[M444](#), [M1815](#), [M2201](#), [M2246](#)
- \if@forced@series [p385](#), [t121](#)
- \if@ignore [A4](#), [A139](#)
- \if@includeinrelease [c68](#), [c71](#), [c98](#), [d418](#)
- \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@nmbrrlist [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](#), [l1478](#), [o99](#), [o658](#),
[s286](#), [s341](#), [s405](#), [s486](#), [s1010](#),
[A30](#), [A226](#), [A247](#), [K52](#), [M990](#),
[M1026](#), [M1626](#), [M1751](#), [N649](#), [N739](#)
- \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 [d327](#), [d344](#), [p403](#),
[p406](#), [p531](#), [p534](#), [t38](#), [t47](#), [N580](#)
- \ifdt@p [B190](#), [B192](#)
- \iff [u494](#)
- \IfFileExists [88](#), [554](#), [a178](#),
[k239](#), [k280](#), [k316](#), [k326](#), [k358](#), [Q484](#)
- \IfFileExists@ [k241](#), [k242](#), [k278](#)
- \iffontchar [H010](#), [H1158](#)
- \ifG@refundefined [z3](#), [z4](#), [z5](#)
- \ifh@ [B76](#), [B114](#), [B123](#)
- \ifin@ [I1494](#),
[l1497](#), [r50](#), [r52](#), [s1](#), [s22](#), [s250](#),
[s352](#), [s354](#), [s415](#), [s428](#), [s498](#),
[s500](#), [s528](#), [s580](#), [s594](#), [s628](#),
[s640](#), [s689](#), [s705](#), [s774](#), [s777](#),
[s798](#), [s828](#), [s831](#), [s894](#), [s897](#),
[s900](#), [s967](#), [s969](#), [s998](#), [t181](#),
[t184](#), [t187](#), [N113](#), [N131](#), [N237](#), [N249](#)
- \ifinner [B231](#),
[B239](#), [B259](#), [B286](#), [I57](#), [I126](#), [I315](#)
- \ifmath@fonts [o204](#), [q176](#)
- \ifmaybe@ic [w82](#), [w91](#)
- \ifnot@nil [q297](#), [q314](#), [q335](#)
- \ifodd [s941](#), [F172](#),
[F192](#), [M21](#), [M138](#), [M610](#), [M668](#),
[M982](#), [M985](#), [M1018](#), [M1021](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

- M1132, M1135, M1294, M1297,
M1574, M1577, M1695, M1698,
M1818, M2039, M2047, I68, I137
- `\IfTargetDateBefore` [N1011](#)
- `\iftc@forced` [x780](#), [x791](#), [x1059](#)
- `\ifv@` [B75](#), [B113](#), [B122](#)
- `\ifvbox` [M319](#), [M376](#), [M423](#), [M502](#), [M518](#)
- `\ignorespaces`
. . . [i49](#), [i117](#), [i136](#), [i148](#), [i159](#),
[i175](#), [i188](#), [i381](#), [k67](#), [k122](#), [l72](#),
[o286](#), [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` [l250](#), [l433](#), [l547](#), [l1092](#)
- `\ij` [l249](#), [l431](#), [l546](#), [l1093](#)
- `\Im` [u308](#)
- `\imath` [u303](#)
- `\in` [u423](#), [u455](#)
- `\in@` [l1492](#),
[l1495](#), [r49](#), [r51](#), [s1](#), [s21](#), [s249](#),
[s351](#), [s353](#), [s411](#), [s424](#), [s497](#),
[s499](#), [s526](#), [s578](#), [s589](#), [s626](#),
[s637](#), [s687](#), [s701](#), [s772](#), [s775](#),
[s795](#), [s826](#), [s829](#), [s891](#), [s895](#),
[s898](#), [s965](#), [s968](#), [s996](#), [t180](#),
[t183](#), [t186](#), [N112](#), [N129](#), [N234](#), [N248](#)
- `\in@%` [p429](#),
[p430](#), [p432](#), [p433](#), [s5](#), [s6](#), [s7](#), [s9](#)
- `\in@false` [s10](#)
- `\in@true` [s12](#)
- `\in_callback` [589](#), [P784](#)
- `\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](#), [d247](#), [d275](#), [d301](#),
[d307](#), [d323](#), [d353](#), [d396](#), [d413](#),
[d422](#), [d442](#), [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](#), [o24](#), [o53](#), [o210](#), [o232](#), [o391](#),
[o402](#), [o410](#), [o442](#), [o450](#), [o464](#),
[o485](#), [o521](#), [o608](#), [o670](#), [p4](#),
[p572](#), [q495](#), [q504](#), [r2](#), [r22](#), [s49](#),
[s78](#), [s138](#), [s169](#), [s575](#), [s623](#), [s669](#),
[s678](#), [s862](#), [s873](#), [t33](#), [t197](#), [t242](#),
[t276](#), [t289](#), [t337](#), [t350](#), [u63](#),
[u81](#), [u100](#), [u106](#), [u616](#), [u628](#),
[w27](#), [w34](#), [x2](#), [x42](#), [x59](#), [x533](#),
[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](#), [N700](#), [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` [469](#), [H176](#),
[J6](#), [J18](#), [L24](#), [L32](#), [M620](#), [M679](#)
- `\indexentry` [J15](#)
- `\inf` [B25](#)
- `\infty` [u310](#)
- `\init@restore@glb@settings`
. [q219](#), [q222](#), [q224](#)
- `\init@restore@version`
. [s62](#), [s91](#), [s108](#), [s123](#), [s124](#)
- `\init@series@setup` [t178](#), [t236](#), [Q544](#)
- `\initcatcodetable` [P90](#)
- `\input` [88](#), [554](#),
[a68](#), [a174](#), [a177](#), [a234](#), [d7](#), [k332](#),
[q16](#), [r106](#), [t396](#), [t406](#), [t416](#), [u10](#),
[u11](#), [u12](#), [u13](#), [u14](#), [u23](#), [u41](#),
[u42](#), [u46](#), [u47](#), [u130](#), [u131](#), [u132](#),
[u133](#), [u648](#), [u649](#), [u650](#), [x1038](#),
[N306](#), [P16](#), [Q97](#), [Q111](#), [Q136](#),
[Q212](#), [Q301](#), [Q489](#), [Q525](#), [Q545](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\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](#), [554](#), [k310](#),
 [k324](#), [k325](#), [k338](#), [k349](#), [k359](#),
 [k375](#), [l1482](#), [o362](#), [t388](#), [t398](#),
 [t408](#), [x754](#), [x1121](#), [N467](#), [O8](#), [Q206](#)
`\inputlineno` [a327](#), [g165](#)
`\insc@unt` [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`
 . [o558](#), [o575](#), [o582](#), [s269](#), [s272](#),
 [s358](#), [s359](#), [s456](#), [s508](#), [s511](#),
 [s518](#), [s533](#), [s534](#), [s541](#), [s981](#), [s983](#)
`\int` [u342](#)
`\interdisplaylinepenalty`
 [i13](#), [B55](#), [B194](#), [B346](#)
`\interfootlinepenalty` [b349](#)
`\interfootnotelinepenalty`
 [b349](#), [i18](#), [I410](#)
`\interlinepenalty` [i11](#), [o634](#), [A227](#),
 [A230](#), [A248](#), [A251](#), [H67](#), [H118](#),
 [H188](#), [H211](#), [M338](#), [M1153](#),
 [M1157](#), [M1319](#), [M1323](#), [I410](#)
`\intertextsep` [M1136](#), [M1140](#), [M1155](#),
 [M1158](#), [M1165](#), [M1298](#), [M1304](#),
 [M1321](#), [M1324](#), [M1333](#), [M2303](#)
`\intop` [u341](#), [u342](#)
`\iota` [u270](#)
`\is@range` [q330](#), [q331](#)
`\ishortstack` [F43](#)
`\itdefault` [p559](#), [t30](#), [u94](#)
`\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` [l444](#), [l801](#),
 [p557](#), [p558](#), [t28](#), [t29](#), [t253](#), [t286](#),
 [t292](#), [w21](#), [x550](#), [G36](#), [G38](#), [I375](#)
- J**
- `\J` [Q190](#), [Q453](#)
- `\j` [l248](#), [l403](#),
 [l545](#), [l773](#), [l1101](#), [l1307](#), [l1387](#), [Q462](#)
`\jmath` [u304](#)
`\Join` [t370](#)
`\joinrel`
 [u465](#), [u472](#), [u474](#), [u476](#), [u478](#),
 [u480](#), [u482](#), [u484](#), [u486](#), [u490](#), [u492](#)
`\jot` [B53](#), [B191](#), [B353](#)
- K**
- `\k` . [l482](#), [l584](#), [l589](#), [l611](#), [l616](#), [l692](#),
 [l693](#), [l751](#), [l752](#), [l806](#), [l808](#), [l813](#),
 [l815](#), [l1193](#), [l1261](#), [l1262](#), [l1279](#),
 [l1280](#), [l1302](#), [l1303](#), [l1304](#), [l1357](#),
 [l1358](#), [l1385](#), [l1386](#), [x190](#), [x208](#)
`\kappa` [u271](#)
`\ker` [B27](#)
`\kernel@ifnextchar` [c76](#),
 [d56](#), [d75](#), [d125](#), [d370](#), [d385](#), [N194](#)
`\kernel@make@fragile`
 [d305](#), [d445](#), [d446](#), [d447](#),
 [d448](#), [d449](#), [d450](#), [d451](#), [d452](#),
 [d453](#), [d454](#), [d455](#), [d456](#), [d457](#),
 [d458](#), [d459](#), [d460](#), [i24](#), [i25](#), [i26](#),
 [i27](#), [i28](#), [l172](#), [l173](#), [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` [l242](#), [l421](#), [l527](#), [l765](#),
 [l1094](#), [N689](#), [N780](#), [N802](#), [Q481](#)
`\l` .. [l251](#), [l423](#), [l548](#), [l774](#), [l1095](#), [Q481](#)
`\l@ngrel@x` .. [d49](#), [d50](#), [d51](#), [d95](#), [d142](#)
`\l@nohyphenation` .. [A223](#), [A337](#), [Q214](#)
`\label` [z32](#), [H176](#), [L24](#), [L32](#), [M619](#), [M678](#)
`\labelformat` ... [330](#), [z39](#), [z44](#), [z50](#), [z56](#)
`\labelsep` .. [C9](#), [C210](#), [C216](#), [G36](#), [G38](#)
`\labelwidth` [C9](#), [C93](#), [C209](#), [C211](#), [C214](#)
`\Lambda` [u294](#)
`\lambda` [u272](#)
`\land` [u362](#), [u364](#)
`\langle` [u588](#)
`\language` [b35](#), [b82](#), [b84](#),
 [b99](#), [k50](#), [A223](#), [A337](#), [M597](#), [O10](#)
`\last@fontshape` [o491](#), [o509](#), [o526](#), [o543](#)
`\lastbox` . [o652](#), [B180](#), [B181](#), [C130](#),
 [C136](#), [C185](#), [H99](#), [H132](#), [M305](#)
`\LastDeclaredEncoding` [o137](#), [o140](#), [Q378](#)
`\lastnamedcs` [d345](#)

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscscmp.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

- `\lastnodetype` . . . o645, o646, o647, o651
- `\lastpenalty` o648, 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, N652, N742
- `\LaTeXe` j13
- `\latexreleaseversion` c1
- `\lbrace` l308, u592
- `\lbrack` b363
- `\lccode` g19, g20, g21,
g22, g23, g24, l140, l1015, A286,
A312, A355, Q157, Q174, Q182,
Q189, Q191, Q192, Q194, Q196,
Q197, Q198, Q199, Q437, Q445,
Q452, Q454, Q455, Q457, Q459
- `\lceil` u596
- `\ldotp` u495, u498, u613
- `\ldots` l322, u499
- `\le` u410, u412
- `\leaders` b435, u334, u552,
u553, u555, u556, E352, F274,
F281, F318, F324, H193, H216
- `\leadsto` t373
- `\leavevmode` b394, b421, b424,
b435, b437, i317, i331, l75,
l184, l289, l290, l393, l422, l426,
l429, l476, l757, l790, w123,
x119, x836, 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,
u618, u620, u622, u624, B115, B141
- `\left` u619, u621, u623, u625, u630,
u631, u632, u633, B154, B160, B182
- `\Leftarrow` u404, u486, u492
- `\leftarrow`
u431, u433, u474, u484, u490, u544
- `\leftarrowfill` u528, u544
- `\lefteqn` B365
- `\leftharpoondown` u447, u461
- `\leftharpoonup` u446
- `\lefthyphenmin` 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` u403
- `\leftrightharpoonrightarrow` u430
- `\leftskip` b416, o629, A180, A184,
A188, A220, A242, C74, D250,
D271, H186, H191, H209, H214
- `\legacyoldstylenums` x1, x52
- `\leq` u408, u410
- `\lfloor` u600
- `\lg` B4
- `\lgroup` u602
- `\lhd` t376
- `\lhook` u471, u472
- `\lim` B6
- `\liminf` B8
- `\limits` u533, u537, B149, B307
- `\limsup` B7
- `\line` g219, F57, F236, F427, F444
- `\linebreak` 71, i9, i26
- `\linepenalty` b307
- `\lineskip` b357, b389,
b424, u452, B187, D252, D272,
E59, E186, F47, F168, M622, M681
- `\lineskiplimit` b358, b389, b426,
b427, u452, u504, B189, B193,
D238, D253, D260, M622, M681
- `\linespread` o287
- `\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` 554, k387
- `\listparindent` C9, C41, C50
- `\ll` u428
- `\llap` C238, C249, D406, D407
- `\lmoustache` u557
- `\ln` B5
- `\lnot` u320, u321
- `\load@onefilewithoptions` N826
- `\LoadClass` 552, N312,
N326, N484, N543, N551, N552
- `\LoadClassWithOptions` 552, N325
- `\LoadFontDefinitionFile` o389,
o415, o416, u21, u27, u28, u29, u33
- `\loccount` P15
- `\log` B3

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\loggingall` [b446](#)
 - `\loggingoutput` [b442](#), [b455](#), [b471](#), [b485](#)
 - `\Longleftarrow` [u486](#)
 - `\longleftarrow` [u483](#)
 - `\Longleftrightarrow` [u492](#), [u494](#)
 - `\longleftrightarrow` [u490](#)
 - `\longmapsto` [u488](#)
 - `\Longrightarrow` [u480](#)
 - `\longrightarrow` [u481](#), [u488](#)
 - `\loop` [a81](#), [b379](#),
[o643](#), [E358](#), [N598](#), [N661](#), [N713](#),
[N751](#), [P150](#), [P159](#), [Q261](#), [Q272](#),
[Q282](#), [Q293](#), [Q323](#), [Q349](#), [Q359](#)
 - `\lor` [u363](#), [u365](#)
 - `\lower` [j2](#), [u452](#), [D169](#),
[F15](#), [F76](#), [F163](#), [F164](#), [F201](#), [F202](#)
 - `\lowerbound` [q340](#), [q341](#), [q352](#)
 - `\lowercase`
[g26](#), [l141](#), [l1016](#), [l1480](#), [o303](#),
[o361](#), [A290](#), [A316](#), [A356](#), [Q477](#)
 - `\lq` [b361](#)
 - `\lrbox` [D99](#)
 - `lrbox` (environment) [376](#)
 - `\ltx@sh@ft` [b430](#),
[l390](#), [l397](#), [l473](#), [l481](#), [l754](#), [l761](#)
 - `\luabytecode` [P193](#)
 - `\luachunk` [P201](#)
 - `\luafunction` [P177](#)
 - `\luafunctionbase` [P275](#)
 - `\luafunctionluafunction` [a18](#), [a23](#)
 - `\luafunctionversion` [a11](#), [P5](#)
- M**
- `\M` [b359](#)
 - `\m@ne` [b39](#)
 - `\m@th` [b410](#), [b422](#), [j13](#),
[u331](#), [u453](#), [u455](#), [u456](#), [u459](#),
[u500](#), [u524](#), [u527](#), [u531](#), [u534](#),
[u541](#), [u544](#), [u551](#), [u554](#), [u636](#),
[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` [d393](#), [k30](#), [k88](#),
[o367](#), [A19](#), [H151](#), [M2](#), [N306](#), [N445](#)
 - `\makeatother` [d393](#), [N306](#), [Q546](#)
 - `\makebox` [376](#), [B252](#), [B278](#), [D3](#)
 - `\makeglossary` [469](#), [k133](#), [J20](#)
 - `\makeindex` [469](#), [k132](#), [J3](#)
 - `\makelabel`
[C45](#), [C97](#), [C205](#), [C218](#), [C238](#), [C249](#)
 - `\MakeLowercase` [Q471](#), [Q483](#)
 - `\makeph@nt` [B101](#), [B103](#)
 - `\MakeRobust` [d246](#),
[d424](#), [d425](#), [d426](#), [d427](#), [d428](#),
[d429](#), [d430](#), [d431](#), [d432](#), [d433](#),
[d434](#), [d435](#), [d436](#), [d437](#), [d438](#),
[d439](#), [s673](#), [s868](#), [F424](#), [F425](#),
[F426](#), [F427](#), [F428](#), [F429](#), [F430](#),
[F431](#), [F432](#), [F433](#), [F434](#), [F435](#)
 - `\makesm@sh` [B131](#), [B133](#)
 - `\maketitle` [438](#)
 - `\MakeUppercase` [z46](#), [Q461](#), [Q461](#)
 - `\mandatory@arg` .. [q368](#), [q455](#), [q459](#),
[q464](#), [q471](#), [q473](#), [q478](#), [q480](#),
[q485](#), [q487](#), [q500](#), [q516](#), [q523](#), [q525](#)
 - `\mapsto` [u438](#)
 - `\mapstochar` [u437](#), [u438](#), [u488](#)
 - `\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` [o589](#), [q260](#), [q266](#), [s53](#),
[s81](#), [s142](#), [s172](#), [w130](#), [w131](#), [w138](#)
 - `\math@egroup` [o589](#),
[q264](#), [q265](#), [w131](#), [w132](#), [w139](#)
 - `\math@fonts` [o559](#), [o564](#),
[q186](#), [q290](#), [s60](#), [s89](#), [s149](#), [s180](#)
 - `\math@fontsfalse`
[j7](#), [l302](#), [l329](#), [l360](#), [l1197](#), [o77](#),
[o206](#), [o216](#), [o239](#), [x124](#), [x584](#), [x841](#)
 - `\math@fontstrue` [o204](#), [o601](#)
 - `\math@version` [o7](#), [o307](#), [o563](#),
[o567](#), [o569](#), [o570](#), [o572](#), [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](#), [t316](#)
 - `\mathaccent` [s590](#), [s638](#), [s672](#), [s682](#)
 - `\mathalpha` . [s760](#), [s939](#), [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](#),

File Key: a=ltmdirchk.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=ltfssaxes.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

u210, u211, u212, u213, u214,
 u215, u216, u217, u218, u219,
 u220, u221, u222, u223, u224,
 u291, u292, u293, u294, u295,
 u296, u297, u298, u299, u300,
 u301, u510, u511, u512, u513,
 u514, u515, u516, u517, u519, u522
 $\backslash\mathbf{b}$ t14, t78, t214, u145
 $\backslash\mathbf{bin}$ s944, u226, u227, u229, u352,
 u353, u354, u355, u358, u359,
 u360, u361, 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, B37
 $\backslash\mathbf{cal}$ u144
 $\backslash\mathbf{char}$
 b422, s701, s745, u329, u330, u611
 $\backslash\mathbf{char@type}$. . s672, s682, s733,
 s736, s745, s761, s866, s877, s940
 $\backslash\mathbf{chardef}$
 b21, b22, b23, b24, b107, b110,
 b111, e3, e4, e5, e6, l70, s736, P214
 $\backslash\mathbf{charzero}$ P214
 $\backslash\mathbf{choice}$ B61
 $\backslash\mathbf{close}$ s947, u225, u234,
 u236, u239, u244, u250, u252,
 u254, u560, u587, u591, u595,
 u599, u605, B43, B46, B49, B52
 $\backslash\mathbf{code}$ s733, u246, u247, u248
 $\backslash\mathbf{dollar}$ l307, u608
 $\backslash\mathbf{ellipsis}$ l321, u613
 $\backslash\mathbf{group}$ b79,
 o14, q257, q263, q269, q270,
 q281, u637, x8, x14, x49, x1048
 $\backslash\mathbf{hexbox}$ b422, t334
 $\backslash\mathbf{indent}$
 . . B379, B391, B403, B431, B441
 $\backslash\mathbf{inner}$ u498, u502, u507, u613
 $\backslash\mathbf{hit}$ p558, t29, u147, u150, u611
 $\backslash\mathbf{normal}$ u143
 $\backslash\mathbf{op}$ s943, u335, u336, u337, u338,
 u339, u340, u341, u343, u344,
 u345, u346, u347, u348, u350,
 u351, u531, u534, 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
 $\backslash\mathbf{open}$ s946,
 u235, u238, u243, u249, u251,
 u253, u558, u589, u593, u597,
 u601, u603, B41, B44, B47, B50
 $\backslash\mathbf{ord}$ s760, s942, u230,
 u237, u240, u245, u257, u258,
 u259, 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, u302, u303, u304, u305,
 u306, u307, u308, u309, u310,
 u311, u312, u313, u314, u315,
 u316, u317, u318, u319, u321,
 u322, u323, u324, u325, u326,
 u327, u328, u518, u520, u521,
 u543, u544, u547, u548, u549,
 u550, u562, u564, u566, u569,
 u571, u585, u607, u608, u609, u610
 $\backslash\mathbf{palette}$ u451,
 u455, u458, B60, B69, B99, B129
 $\backslash\mathbf{paragraph}$ l310, m134, m146, u608
 $\backslash\mathbf{ph@nt}$ B99, B105
 $\backslash\mathbf{punct}$
 s948, u228, u232, u495, u496, u497
 $\backslash\mathbf{rel}$. . s945, u231, u233, u241,
 u242, u255, u256, u332, u392,
 u393, u394, u395, u396, u397,
 u398, u399, u400, u401, u402,
 u403, u404, u405, u408, u409,
 u412, u413, u414, u415, u416,
 u417, u418, u419, u420, u421,
 u422, u423, u424, u426, u427,
 u428, u429, u430, u431, u432,
 u435, u436, u437, u439, u440,
 u441, u442, u443, u444, u445,
 u446, u447, u448, u449, u451,
 u455, u458, u465, u467, u470,
 u471, u473, u476, u478, u573,
 u575, u577, u579, u581, u583,
 B42, B45, B48, B51, B149, B307
 $\backslash\mathbf{ring}$ u522
 $\backslash\mathbf{rm}$ t5, t98, t220, u142
 $\backslash\mathbf{section}$. . l311, m133, m145, u608
 $\backslash\mathbf{sf}$ t8, t103, t223, u146, u149
 $\backslash\mathbf{sm@sh}$ B129, B135
 $\backslash\mathbf{sterling}$ l319, u608
 $\backslash\mathbf{strut}$ B84, B93, B158, B159
 $\backslash\mathbf{surround}$ b410
 $\backslash\mathbf{symbol}$ s738
 $\backslash\mathbf{t}$ t11, t108, t226, u148
 $\backslash\mathbf{underscore}$ u608
 $\backslash\mathbf{version}$ o307, t313, t315
 $\backslash\mathbf{matrix}$ B156, B160, B167

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\max` B22
 - `\maxdeadcycles` M7
 - `\maxdepth` b331,
i237, k57, k112, M92, M169,
M170, M506, M514, M546,
M715, M724, M764, M991, Q85
 - `\maxdimen` b296, b332, b333,
b389, b427, b443, b454, b470,
b485, o616, o626, o661, o676,
q338, q391, u452, F247, F292,
F331, M291, M1853, M1873,
M1878, M2165, M2205, M2206,
M2208, N823, N864, N873, Q89
 - `\maybe@ic` w63, w64, w83
 - `\maybe@ic@` w83
 - `\maybe@icfalse` w97
 - `\maybe@ictrue` w87
 - `\maybe@load@fontshape`
. l71, p401, p423, p529, p584, t142
 - `\mb@b` D50, D60
 - `\mb@l` D50, D54, D59, F48, F52
 - `\mb@r` D50, D54, D59, F48, F52
 - `\mb@t` D51, D58
 - `\mbox` 376, b422, j13, l293, t330, u500,
D11, D20, D24, F20, I381, I389
 - `\mddef@ult` t74, t170
 - `\mddefault` t18, t74, t93, t218, u92, u114
 - `\mdseries`
t16, t17, t87, t216, t217, t333, w20
 - `\mdseries@rm` t60, t90, t207
 - `\mdseries@sf` t60, t91, t208
 - `\mdseries@tt` t60, t92, t209
 - `\meaning` a219, a228, a323, d203, d262,
d289, d391, s412, s425, s526,
s591, s638, s702, s796, s892, s996
 - `\medbreak` b404, d431, d452
 - `\medmuskip` u639, B36, B38, B202
 - `\medskip` b407, i310
 - `\medskipamount` b406, i311, i313
 - `\merge@font@series` .. p387, p388, p581
 - `\merge@font@series@` . p390, p395, p582
 - `\merge@font@shape` .. p515, p517, p589
 - `\merge@font@shape@` .. p519, p524, p590
 - `\MessageBreak` d179,
d252, d279, g3, g6, g13, g33,
g46, g60, g73, g171, g173, g179,
g186, l161, l971, l1485, l1488,
o34, o35, o507, o541, p419,
q20, q21, q67, q88, q281, q432,
q452, q484, q500, q515, q528,
r31, r33, s367, s376, s514, t55,
w144, x23, x100, x102, x121,
x760, x762, x763, x764, x766,
x768, x769, x770, x771, x772,
x822, x824, x831, x838, x1053,
A23, M578, M1960, M1997,
N145, N336, N347, N349, N351,
N362, N452, N453, N455, N456,
N457, N459, N461, N479, N480,
N481, N482, N528, N545, N546,
N620, N639, N678, N734, N768,
N877, N879, N961, N964, N977,
N979, Q230, Q231, Q232, Q234
 - `\mho` t369
 - `\mid` u396
 - `\min` B23
 - `\minipage` D281
 - `minipage (environment)` 377
 - `\mit` t419
 - `\mkern` u329, u332,
u334, u456, u465, u507, u508,
u509, u539, u540, u541, u542,
u543, u544, u545, u546, B36,
B37, B40, B73, B74, H194, H217
 - `\mlist_to_hlist` P833
 - `\models` u478
 - `\module_error` 588, P331
 - `\module_info` 588, P331
 - `\module_warning` 588, P331
 - `\modules` P284
 - `\month` a185, c17, N656, N746
 - `\moveright` M625, M684
 - `\mp` u383
 - `\mscount` E355
 - `\mskip` i336,
B36, B38, B201, B202, B203, B204
 - `\mu` u273
 - `\mubyte` Q248
 - `\multicolumn` E209
 - `\multiput` F25, F29, F429, F446
 - `\multispan` E209, E355
 - `\muskip` . b29, b55, b93, u539, u540, P32
 - `\muskipdef` b55, b93, P215
 - `\muskipzero` P215
- N
- `\n` P318, P320, P327, P329,
P456, P562, P588, P614, P660,
P682, P701, P709, P710, P730,
P743, P750, P751, P758, P770
 - `\n@space` u619, u621, u623,
u625, u630, u631, u632, u633, u636
 - `\nabla` u313
 - `\narrower` b415
 - `\natural` u323
 - `\ncallback` P665
 - `\ndefault` P670, P674

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\ne` [u407](#)
- `\nearrow` [u399](#)
- `\NeedsTeXFormat` [q12](#), [x731](#), [N341](#), [N1023](#)
- `\neg` [u319](#), [u320](#)
- `\negthinspace` [i361](#), [i370](#)
- `\neq` [u406](#)
- `\newcommand`
 [d52](#), [d53](#), [d106](#), [d140](#), [d159](#), [d214](#)
- `\newenvironment` ... [d121](#), [d122](#), [d134](#)
- `\newfontshape` [r2](#), [r4](#), [r22](#), [r24](#)
- `\newmathalphabet` ... [s409](#), [s430](#), [s441](#)
- `\newmathgroup`
 [b78](#), [b80](#), [b98](#), [b100](#), [o14](#), [s289](#), [P25](#)
- `\newmathversion` ... [s20](#), [s246](#), [s248](#)
- `\newsymbolfont` [s290](#), [s322](#)
- `\new_attribute` [586](#), [P397](#)
- `\new_bytecode` [586](#), [P431](#)
- `\new_chunkname` [586](#), [P444](#)
- `\new_luafunction` [586](#), [P460](#)
- `\new_whatsit` [586](#), [P419](#)
- `\newattribute` [585](#), [P74](#), [P225](#)
- `\newbox` [b47](#), [b301](#), [b412](#),
 [e13](#), [A283](#), [B66](#), [C27](#), [D70](#),
 [E16](#), [E17](#), [E18](#), [E319](#), [F6](#), [F356](#),
 [F361](#), [M86](#), [M120](#), [M121](#), [M122](#)
- `\newcatcodetable` [585](#),
 [P84](#), [P93](#), [P94](#), [P120](#), [P121](#), [P229](#)
- `\newcommand` [37](#), [d52](#), [l4](#), [p449](#), [p453](#),
 [p457](#), [t35](#), [u51](#), [u52](#), [u53](#), [u54](#),
 [u56](#), [u57](#), [u59](#), [u60](#), [u92](#), [u93](#),
 [u94](#), [u95](#), [u96](#), [u97](#), [u113](#), [u114](#),
 [u116](#), [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](#), [N820](#), [Q3](#), [Q4](#),
 [Q5](#), [Q77](#), [I3](#), [I267](#), [I268](#), [I269](#), [I270](#)
- `\newcounter` [146](#), [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](#), [N654](#), [N744](#)
- `\newfam` [b47](#), [o16](#), [P36](#)
- `\newfont` [t317](#)
- `\newgroup` [s47](#)
- `\newhelp` [b295](#)
- `\newif` [c68](#), [d143](#), [d418](#), [e9](#), [k7](#),
 [k8](#), [o204](#), [p385](#), [s15](#), [w82](#), [x780](#),
 [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` [153](#), [n3](#)
- `\newline` [i79](#)
- `\newlinechar` [a72](#), [d5](#)
- `\newluabytecode` [585](#), [P189](#), [P239](#)
- `\newluachunkname` ... [586](#), [P197](#), [P241](#)
- `\newluafunction`
 [585](#), [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` [376](#), [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](#), [x194](#), [x212](#), [x615](#), [x924](#), [x925](#)
- `\newtoks` [b47](#),
 [b295](#), [e16](#), [o317](#), [o318](#), [p439](#), [q201](#)
- `\newwhatsit` [585](#), [P181](#), [P237](#)
- `\newwrite` [b47](#), [k4](#), [k5](#), [k6](#), [H154](#), [J4](#), [J21](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

`\newXeTeXintercharclass` [Q21](#)
`\nfss@catcodes` [o19](#), [o120](#), [o358](#), [o359](#),
[o366](#), [o413](#), [u40](#), [u45](#), [u129](#), [M3](#)
`\nfss@text` [l315](#), [l317](#), [t330](#), [w5](#), [w122](#), [z13](#)
`\NG` [l528](#), [l1096](#), [Q481](#)
`\ng` [l549](#), [l1097](#), [Q481](#)
`\ni` [u424](#), [u425](#)
`\no@alphabet@error` . [o4](#), [s268](#), [s270](#),
[s446](#), [s447](#), [s461](#), [s470](#), [s556](#), [s557](#)
`\noaccents@` [o604](#), [u123](#)
`\noalign` [u333](#),
[u525](#), [u528](#), [u531](#), [u532](#), [u536](#),
[u537](#), [B158](#), [B159](#), [B175](#), [B178](#),
[B192](#), [B353](#), [E208](#), [E335](#), [E354](#), [F55](#)
`\nobreak` [b392](#), [b395](#), [b397](#),
[d432](#), [d453](#), [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` [471](#), [K39](#)
`\nocorr` [w43](#), [w58](#), [w62](#), [w65](#)
`\nocorrlist` [w89](#), [w121](#)
`\nofiles` [88](#), [k127](#)
`\noindent` [o639](#), [o665](#), [H139](#)
`\nointerlineskip`
. [b387](#), [u333](#), [u525](#), [u528](#),
[u532](#), [u536](#), [B251](#), [B277](#), [F272](#),
[F275](#), [F316](#), [F318](#), [M1842](#), [M1850](#)
`\nolimits` . [u342](#), [u349](#), [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` [o583](#), [o585](#),
[s72](#), [s101](#), [s117](#), [s163](#), [s194](#), [s1018](#)
`\nonfrenchspacing`
[b353](#), [b541](#), [d433](#), [d454](#), [k46](#), [k104](#)
`\nonscript` [B36](#), [B38](#)
`\nonumber` [B340](#), [B363](#), [B364](#)
`\nopagebreak` [71](#), [i7](#), [i25](#)
`\noprotusion` [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` [o617](#), [o677](#), [t335](#),
[t351](#), [t353](#), [t359](#), [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](#)
`\normalshape` [p544](#), [p591](#), [p598](#)
`\normalsize` [k40](#), [k98](#), [w142](#),
[M617](#), [M676](#), [N5](#), [I23](#), [I176](#), [I348](#)
`\not` [u332](#), [u406](#), [u407](#), [u429](#)
`\not@base` [t365](#),
[t369](#), [t370](#), [t371](#), [t372](#), [t373](#),
[t374](#), [t375](#), [t376](#), [t377](#), [t378](#), [t379](#)
`\not@math@alphabet` . . . [p447](#), [p451](#),
[p455](#), [p545](#), [p549](#), [p552](#), [p555](#),
[p558](#), [p561](#), [p564](#), [p567](#), [t5](#), [t8](#),
[t11](#), [t14](#), [t17](#), [t20](#), [t23](#), [t26](#),
[t29](#), [t78](#), [t88](#), [t98](#), [t103](#), [t108](#),
[t214](#), [t217](#), [t220](#), [t223](#), [t226](#), [t296](#)
`\notin` [u455](#)
`\nu` [u274](#)
`\null` [b371](#), [l327](#), [l363](#),
[l483](#), [l486](#), [l828](#), [l831](#), [l1195](#), [z17](#),
[A227](#), [A248](#), [A333](#), [A343](#), [B112](#),
[B121](#), [B156](#), [B185](#), [H191](#), [H214](#)
`\nulldelimiterspace` [b335](#), [u636](#)
`\nullfont` [A88](#)
`\number` [a86](#), [c48](#),
[c56](#), [d2](#), [d89](#), [m108](#), [o567](#), [o570](#),
[q393](#), [s64](#), [s93](#), [s113](#), [s128](#), [s153](#),
[s184](#), [t327](#), [N559](#), [N656](#), [N746](#), [P105](#)
`\numberline` [H72](#), [H82](#), [H227](#), [I17](#)
`\numexpr` . . [b189](#), [b205](#), [b215](#), [b235](#),
[d334](#), [l994](#), [M36](#), [P82](#), [P105](#), [P157](#)
`\unknown` [P687](#)
`\narrow` [u401](#)

O

File Key: a=ltmdirchk.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=ltfssaxes.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

- \of B67, B310
 - \offinterlineskip b387
 - \oint u349
 - \ointop u348, u349
 - \oldstylenums x1, x43, x45,
x303, x304, x305, x306, x307,
x308, x309, x310, x311, x312, x1045
 - \Omega u301
 - \omega u284
 - \ominus u381
 - \omit B178,
B179, E345, E348, E355, E359
 - \on@line g8, g15, g165,
t122, t125, A97, A114, D105, N478
 - \onecolumn M141
 - \OnlyDescription q5, v3
 - \ooalign b424,
l327, l357, l394, l477, l483,
l485, l496, l512, l725, l758, l828,
l831, l877, l1195, t332, u456, u459
 - \openup B186, B191
 - \operator@font u637, 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 u382
 - \optional@arg
q369, q448, q450, q522, q525
 - \OptionNotUsed N218, N225, N493
 - \oslash u379
 - \OT l372
 - \otimes u380
 - \outer P19, P36
 - \outer@nobreak .. I181, I251, I255, I342
 - \outerparskip C1
 - \output M256
 - \outputpenalty M258, M272,
M295, M298, M299, M334,
M1159, M1160, M1325, M1328
 - \oval F236, F239, F430, F447
 - \over u463, B149, B308
 - \overbrace u530
 - \overfullrule b330, L69
 - \overleftarrow u527
 - \overrightarrow u524
 - \owns u425, u426
- P**
- \P l310
 - \p@ b298
 - \p@equation B318, B438
 - \p@selectfont q117
 - \PackageError c72, c101,
c110, g84, l1483, x736, x786, x830
 - \PackageInfo g84, x738,
x755, x760, x776, x777, x837, x1122
 - \PackageWarning g84, x737, x787, x1051
 - \PackageWarningNoLine g84, l969, M1959
 - \pagebreak 71, i6, i7, i22, i24
 - \pagegoal M1880, M1887
 - \pagenumbering 328, 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, o642,
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 u395
 - \parbox 376, D189
 - \parboxrestore D277
 - \parfillskip b345, o616, o631,
o676, 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 u309
 - \partopsep B445, C1, C61
 - \PassOptionsToClass 552, N195
 - \PassOptionsToPackage 552, N195
 - \patch@level c1, c36, c41, c43,
c45, c48, c56, Q491, Q503, Q505
 - \patterns l205

File Key: a=ltmdirchk.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=ltfssaxes.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

- \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 u441
- \phant B81, B82, B83, B97
- \phantom [B75](#)
- \Phi u299
- \phi u281
- \Pi u296
- \pi u276
- \pickup@font I181, o195, o324,
o518, o552, q122, q285, q287, q289
- \pictur@ [F8](#)
- \picture [F8](#)
- \pkgcls@arg N837, N960
- \pkgcls@candidate [N824](#), N839,
N915, N919, N923, N991, N994
- \pkgcls@debug [N814](#), N830, N831,
N832, N833, N834, N891, N892,
N893, N894, N903, N908, N926,
N935, N950, N984, N985, N986
- \pkgcls@innerdate
. [N819](#), N864, N867, N873, N1012
- \pkgcls@mindate
 N844, N853, N869, N874
- \pkgcls@name N836, N879
- \pkgcls@parse@date@arg N838, [N849](#)
- \pkgcls@parse@date@arg@ N855, N858
- \pkgcls@parse@date@arg@version
 N865, N886
- \pkgcls@releasedate
 [N824](#), N920, N924, N995
- \pkgcls@rollbackdate@error
 N916, [N974](#), N992
- \pkgcls@show@selection
 N943, [N948](#), N998, N1003
- \pkgcls@targetdate
 [N819](#), N851, N859,
N862, N863, N867, N875, N876,
N889, N897, N912, N914, N944,
N955, N957, N982, N988, N990
- \pkgcls@targetlabel
 [N819](#), N852, N872, N887,
N899, N931, N964, N1002, N1005
- \pkgcls@use@this@release N900,
N917, N919, N932, [N942](#), N994
- \pm u384
- \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 u415
- \preceq u418
- \predisplaypenalty
 b316, B387, B399, B425
- \preload@sizes r11, [r94](#)
- \prepare@family@series@update
 t99, t104, t109, [t119](#), t233
- \pretolerance b303, o618, o633, o678
- \prevdepth b387, b391, b392, i237,
i238, i295, i300, B192, M167,
M169, M172, I196, I198, I218, I220
- \prim@s B207, B209, B221
- \prime u247, u311, B210
- \prime@s [B208](#)
- \process@table k38, k96, [s200](#)
- \ProcessOptions I1504,
q71, x753, x790, [N226](#), N269, N547
- \ProcessOptions* [N226](#)
- \prod u343
- \propto u392
- \protect d77, d195, d209, d218, d223,
d226, d227, d229, d230, d235,
d236, d241, d244, [d245](#), d267,
d294, g197, g199, g200, g206,
g212, g219, g227, g230, g236,
k139, l26, l32, l51, l55, l209,
l217, s475, s1023, t320, w143,
z12, A103, A106, A121, A131,
E240, H12, H72, H82, H164,
H171, K5, M596, M655, Q254, I17
- \protected i56,
i364, l308, l309, m160, p544,
p548, p551, p554, p557, p560,
p563, p566, s865, t272, A47, A157
- \protected@edef [d228](#),
m158, t245, 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
 [d228](#), H11, N155, I400, I424, I440
- \provide@command [d153](#), [d154](#)

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscnp.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

- \providecommand
 d153, 16, 1964, x723, x724, M1970
 \provides_module 588, P285
 \ProvidesClass 552, N182
 \ProvidesFile
 a89, u659, u661, u662, u663, N184
 \ProvidesPackage 552, q13,
 x729, x758, N141, N182, N1024
 \ProvideTextCommand l3, l60
 \ProvideTextCommandDefault l57
 \ps@empty L10, Q91
 \ps@plain L13
 \Psi u300
 \psi u283
 \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 . N609, N611, N616, N644
 \qbezier 411, F368, F432, F449
 \qbeziermax F367, F389
 \qqquad i375
 \quad ... i375, B155, B157, B177, H111
 \quote@name k236, k274
 \quote@name k236, k260, k262, k273, N609
 \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, x191, x209
 \r@t B66
 \radical s889, s892, s922
 \raggedbottom L53
 \raggedleft A186, A197, A204
 \raggedright A182, A196, A202
 \raise l327, l359, l427, l430,
 l726, l791, l878, l1195, t333,
 u459, u507, u509, B73, D386,
 D395, F22, F32, F75, F163,
 F238, F266, F310, F338, F416
 \raisebox 377, l1164, D363
 \rangle u586
 \rbrace l309, u590
 \rbrack b363
 \rceil u594
 \Re u307
 \Ref 330, z39, z45, z50, z57
- \ref z10, z45
 \refstepcounter
 146, z39, z40, z50, z52,
 B304, B427, C202, G27, H59, I9
 \registernumber 587, P376
 \reinstall@nfss@defs p547, p600, Q544
 \Relbar u470, u478, u480, u486
 \relbar u467, u482, u484
 \relpenalty b311
 \rem@pt o300
 \remove@angles q301, q324
 \remove@nil s36
 \remove@star q301, q307
 \remove@tlig l983,
 l985, l987, l994, l1030, l1032, l1034
 \remove@to@nnil o299, q301, q327, q440
 \remove_from_callback 588, P741
 \removelastskip b403, b405, b407, b409
 \renew@command . d99, d100, d160, d168
 \renew@environment d127, d128
 \renewcommand 37, d99, u66,
 u68, u70, u71, u73, u75, u77,
 u78, u84, u86, u88, u89, u102,
 u103, u109, u110, B375, B395, B416
 \renewenvironment 37, d127, B424, B436
 \repeat a81, a83, b379,
 o659, E358, N602, N665, N717,
 N755, P154, P164, Q265, Q276,
 Q286, Q297, Q327, Q353, Q363
 \requestedLaTeXdate
 N818, N851, N871, N957
 \requestedpatchdate N881, N958
 \RequirePackage 552, M1967,
 N302, N309, N330, N543, P22
 \RequirePackageWithOptions 552, 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,
 d193, d201, d205, d260, d264,
 d287, d291, d367, d376, 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,

File Key: a=ltmdirchk.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=ltfssaxes.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

197, 1107, 1110, 1119, 1138, 1143,
 1977, 1981, 11008, 11017, o30,
 o32, o41, o44, o47, o63, o66,
 o69, o105, o108, o110, o147,
 o151, o360, o363, o490, o491,
 o506, o509, o514, o525, o526,
 o539, o543, o548, o575, o578,
 o579, o587, p397, p398, p402,
 p403, p418, p419, p525, p526,
 p530, p531, q150, q152, q154,
 q164, q166, q169, q298, q299,
 q312, q313, r53, r57, s356, s365,
 s367, s411, s414, s424, s427,
 s525, s527, s589, s593, s637,
 s639, s700, s703, s795, s797,
 s891, s893, s995, s997, s1013,
 s1015, s1016, s1021, t36, t37,
 t151, t152, 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, N579, N580, N581, N583,
 N625, N626, N628, N632, N841,
 N846, N898, N899, N930, N931,
 N1001, N1002, N1026, N1028,
 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,
 d202, d203, d205, d261, d262,
 d264, d288, d289, d291, d368,
 d378, f33, f34, f37, i320, i321,
 i328, k186, k188, k258, k260,
 k262, k297, k299, k301, k402,
 k408, 190, 197, 1112, 1119, 1980,
 1981, 11017, 11026, 11028, o31,
 o32, o38, o95, o97, o150, o151,
 o576, o587, p417, p418, p420,
 r47, r54, r71, r73, s282, s284,
 s337, s339, s364, s365, s366,
 s401, s403, s482, s484, s529,
 s530, s531, s538, s698, s702,
 s704, t158, t163, t165, t166,
 t170, t171, 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, N627, N628, N631, N632,
 N667, N668, N670, N696, N757,
 N758, N760, N787, Q162, Q164,
 Q168, Q393, Q395, Q399, Q464,
 Q473, Q526, I43, I44, I112, I113
 \reserved@c a123,
 a128, d373, d376, d378, d381,
 k391, k392, o96, o97, o577,
 o580, r48, r55, r61, r68, s33,
 s37, s283, s284, s338, s339, s402,
 s403, s483, s484, s506, s515,
 s530, s544, s785, s802, s811,
 s839, s850, s908, s921, s923,
 t160, t163, t168, t169, t172,
 t173, w67, w69, w76, N610,
 N615, N616, N636, N644, N650,
 N672, N679, N728, N729, N730,
 N740, N762, N769, N797, Q166,
 Q171, Q179, Q387, Q397, Q418,
 Q419, Q420, Q422, Q423, Q424,
 Q425, Q426, Q434, Q442, Q528
 \reserved@d a126, a129,
 d366, d375, k390, k392, r61,
 r68, r70, r74, s793, s802, s811,
 s847, s850, s916, s921, s925, Q529
 \reserved@e
 i57, i59, i69, i71, i83, i89, r39,
 r45, r70, r73, r74, s34, s39, Q530
 \reserved@f i58, i59,
 i70, i71, i89, 11479, 11480, 11481,
 11482, 11484, 11491, o190, o192,
 o198, o199, q336, q347, q351,
 q355, q361, q364, q403, q440,
 q443, r27, r38, r45, r71, r73, Q531
 \reset@font
 .. t335, t359, z13, D327, K20,
 L14, M616, M675, I175, I347, I409
 \restglb@settings q222, q232
 \restore@mathversion
 s107, s110, s125, s133

File Key: a=ltmdirchk.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=ltfssaxes.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

- [\restore@protect](#) [d228](#)
[\restorecr](#) [i378](#)
[\reversemarginpar](#) [I363](#)
[\rfloor](#) [u598](#)
[\rgroup](#) [u602](#)
[\rhd](#) [t378](#)
[\rho](#) [u277](#)
[\rhook](#) [u473, u474](#)
[\right](#) [u619, u621, u623, u625, u630,](#)
[u631, u632, u633, B155, B160, B184](#)
[\Rightarrow](#) [u405, u480, u492](#)
[\rightarrow](#) [u432,](#)
[u434, u438, u472, u482, u490, u543](#)
[\rightarrowfill](#) [u525, u541](#)
[\rightharpoondown](#) [u449](#)
[\rightharpoonup](#) [u448, u460](#)
[\rightthypenmin](#) [O11](#)
[\rightleftharpoons](#) [u458](#)
[\rightline](#) [D402](#)
[\rightmargin](#) [C9, C40, C51](#)
[\rightmark](#) [L48](#)
[\rightskip](#) [b417, o630,](#)
[A180, A183, A188, A220, A242,](#)
[C75, D250, D271, H186, H209](#)
[\rlap](#) [l427,](#)
[l430, l791, B365, B376, D406, E69](#)
[\rlh@](#) [u458, u459](#)
[\rmdef@ult](#) [t70, t80, t90, t189, x27, x105](#)
[\rmdefault](#) [t6, t70,](#)
[t99, t180, t221, u50, u113, x7, x48](#)
[\rmfamily](#)
[t4, t5, t97, t189, t219, t220, w15](#)
[\rmoustache](#) [u559](#)
[\rmsubstdefault](#) [u18, u30, x28, x39, x106](#)
[\Roman](#) [146, m104](#)
[\roman](#) [146, m103](#)
[\romannumeral](#) [m109,](#)
[m110, A120, C43, C234, C245](#)
[\root](#) [B66, B310](#)
[\rootbox](#) [B66](#)
[\rq](#) [b361](#)
[\rule](#) [377, D334, D337, I419](#)
- S**
- [\S](#) [l311](#)
[\sfct@](#) [q380, q444](#)
[\sfct@alias](#) [q506](#)
[\sfct@fixed](#) [q519](#)
[\sfct@gen](#) [q456](#)
[\sfct@genb](#) [q461](#)
[\sfct@sgen](#) [q456](#)
[\sfct@sgenb](#) [q461](#)
[\sfct@sub](#) [q468](#)
[\sfct@subf](#) [q511](#)
[\samepage](#) [71, i11, i28](#)
[\savebox](#) [376, D71](#)
[\savecatcodetable](#) . [P117, P168, P170](#)
[\saved@space@catcode](#) ... [Q251, Q320](#)
[\sb](#) [B199](#)
[\sbox](#) [376, b411, j4, l495, l511,](#)
[C205, D77, D84, D88, D93, D98](#)
[\scan@fontshape](#) [r7, r40, r43](#)
[\scanfontshape](#) [r6, r26, r37](#)
[\scdefault](#) [p556, t27, u94](#)
[\scriptfont](#) [q292](#)
[\scriptfont@name](#) [q287, q292](#)
[\scriptscriptfont](#) [q293](#)
[\scriptscriptstyle](#) [B65, B68](#)
[\scriptspace](#) [b336](#)
[\scriptstyle](#) [u331, B64](#)
[\scshape](#) [l300, p554, p555, t25, t26, w23](#)
[\searrow](#) [u400](#)
[\sec](#) [B20](#)
[\secdef](#) [H142](#)
[\sectionmark](#) [H143](#)
[\select@group](#) [o560, o579, s48, s236,](#)
[s273, s411, s464, s473, s511, s543](#)
[\selectfont](#) [j7, l302, l329,](#)
[l360, l446, l803, l1163, l1197,](#)
[l1505, o285, p448, p452, p456,](#)
[p546, p550, p553, p556, p559,](#)
[p562, p565, p568, q112, t6, t9,](#)
[t12, t15, t18, t21, t24, t27, t30,](#)
[t85, t95, t101, t106, t111, t215,](#)
[t218, t221, t224, t227, t263,](#)
[t323, t345, x36, x115, x125,](#)
[x552, x584, x825, x842, l379, l387](#)
[\series@change@debug](#) [t115,](#)
[t122, t125, t136, t139, t143,](#)
[t154, t162, t165, t168, t171, t173](#)
[\series@check@toks](#) .. [p430, p432, p439](#)
[\series@drop@one@m](#) .. [p436, p440, p587](#)
[\series@maybe@drop@one@m](#)
[o31, p426, p428, p586](#)
[\seriesdefault](#)
[l1506, s239, t342, t356, u113](#)
[\set@mathdelimiter](#) [s848, s882](#)
[\set@color](#) [D62](#)
[\set@curr@file](#) [k152,](#)
[k160, k222, k240, k271, k276, N608](#)
[\set@display@protect](#)
[d3, d226, g7, g14, g34, g61, N163](#)
[\set@fontsize](#) .. [o288, o290, q119, q132](#)
[\set@mathaccent](#)
[s598, s606, s641, s649, s667](#)
[\set@mathchar](#) [s722, s732](#)
[\set@mathdelimiter](#) ... [s799, s808, s860](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\set@mathradical` s244, s918
- `\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@target@series`
 - p399, p404, p407, p410, [p424](#), p585
- `\set@typeset@protect`
 - [d226](#), d245, E185,
 - E211, M603, M605, M661, M663
- `\setattribute` 586, [P82](#), P226
- `\setcounter`
 - 146, k215, [m2](#), m37, C225,
 - M2290, M2293, M2296, M2300
- `\setlength`
 - 153, [n4](#), B443, B448, B449,
 - B450, D43, D159, D220, D223,
 - D296, D353, D354, D355, D384,
 - D385, D392, D393, D394, E164,
 - E360, M2306, M2307, M2308,
 - M2311, M2312, M2316, M2317,
 - M2318, M2322, M2323, M2324
- `\SetMathAlphabet`
 - o11, r140, r141, [s480](#), u149, u150
- `\SetMathAlphabet@` s418, s487, [s496](#)
- `\setminus` u387
- `\setrangecatcode` P96, P104, P113, P114
- `\SetSymbolFont` s335, u139, u140, u141
- `\SetSymbolFont@` s308, s342, [s350](#)
- `\settodepth` 153, [n17](#)
- `\settoheight` 153, [n17](#)
- `\settowidth` 153, [n17](#)
- `\sf@size` j6, l302, o224, o243,
 - o599, q282, q286, x584, I381, I389
- `\sfcode` b353, b354, b355, b356,
 - b440, i326, k43, k101, Q183, Q446
- `\sfdef@ult` t71, t81, t91, t190, x29, x107
- `\sfdefault` t9, t71, t104, t183, t224, [u50](#)
- `\sffamily`
 - t7, t8, [t102](#), t190, t222, t223, w16
- `\sfsubstdefault` u19, u31, x30, x108
- `\sh@ft` [b428](#)
- `\shapedefault`
 - i1506, p546, s240, t343, t357, [u113](#)
- `\sharp` u324
- `\shipout` M602, M660
- `\shortstack` F43, F433, F450
- `\showboxbreadth`
 - b325, b443, b496, b513, b529
- `\showboxdepth` b326, b443,
 - b495, b512, b530, o618, o662, o679
- `\showhyphens` [o607](#)
- `\showoutput` [b442](#)
- `\showoverfull` [b441](#), b444, b478, b486
- `\Sigma` u297
- `\sigma` u278
- `\sim` u439, u451
- `\simeq` u440
- `\sin` B9
- `\sinh` B11
- `\sixt@n` a71,
 - [b16](#), b64, b66, b96, b97, b98,
 - o14, s84, s175, s584, s586, s632,
 - s634, s693, s695, s741, s743,
 - s781, s783, s789, s791, s835,
 - s837, s843, s845, s904, s906,
 - s912, s914, 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` u538
- `\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](#), d436, d457
- `\sldefault` p553, t24, [u94](#)
- `\sloppy` D255, D274, [L57](#), L62
- `\sloppypar` L62
- `sloppypar (environment)` [L62](#)
- `\slshape` l437, l794,
 - p551, p552, t22, t23, w22, x543
- `\smallbreak` [b404](#), d437, d458
- `\smallint` u351
- `\smallskip` b405, [i310](#)
- `\smallskipamount` b404, i310, [i313](#)
- `\smash` u467, u541, u542, u545, u546, [B126](#)
- `\smile` u444
- `\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` u328
- `\span` [E359](#)
- `\split@name`
 - [o328](#), o340, o454, o469, q473, q487
- `\splitfirstmark` M2211
- `\splitmaxdepth` b332, M2205, I412

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\splittopskip` b344, I411
`\sqcap` u370
`\sqcup` u371
`\sqrt` B309
`\sqrtsign` u523, B71, B309
`\sqsubset` t374
`\sqsubseteq` u393
`\sqsupset` t375
`\sqsupseteq` u394
`\SS` l304, l531, l1108, Q481
`\ss` . l254, l406, l554, l778, l1083, Q481
`\sscdefault` p454, p568, p597
`\sscshape` p454, p566, p567, p596, w31
`\ssf@size` l328, l360,
l1196, o225, o244, o600, q282, q288
`\stackrel` B307
`\star` u391
`\stepcounter` l46, m17, m27,
o572, s48, z40, z52, B317, B360,
B437, M646, M705, I399, I423
`\stop` A86
`\storedpar` P156, P161
`\stretch` i356
`\strip@prefix` a111, a228,
a323, d203, d262, d289, d390, o557
`\strip@pt` b432,
o216, o222, o223, o224, o225,
o238, o242, o300, o599, o600, q134
`\strut` b412, d438, d459, 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` u420
`\subseteq` u422
`\subst@correction` o85, o91
`\subst@fontshape` r8, r80
`\subst@size` q419
`\subsubsectionmark` H143
`\succ` u414
`\succeq` u417
`\sum` u344
`\sup` B24
`\suppressfloats` M1972
`\supset` u419
`\supseteq` u421
`\surd` u330
`\sw@slant` w91, w101
`\swarrow` u402
`\swdefault` p450, p565, p595
`\swshape` . p450, p563, p564, p594, w30
`\symbol` l162, t317
`\symletters` x8, x14, x49, x1048
`\symoperators` u637
- T**
- `\T` g23, l335,
l337, l339, l341, l343, l345, l347,
l349, l351, l374, N798, N802, N803
`\t` l282, l735, l843, x177, x178,
x192, x194, x195, x207, x210,
x212, x598, x768, x1035, x1037
`\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`
..... t128, t152, t159, t161
`\target@series@value`
..... t127, t135, t138, t140,
t144, t145, t166, t169, t170, t172
`\tau` u279
`\tc@check@accent`
x130, x180, x181, x182, x183,
x184, x185, x186, x187, x188,
x189, x190, x191, x192, x193,
x194, x195, x847, x923, x925, x927
`\tc@check@symbol` x130, x214,
x215, x216, x217, 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, x264, x265,
x266, x267, x268, x269, x270,
x271, x272, x273, x274, x275,
x276, x277, x278, x279, x280,
x281, x282, x283, x284, x287,
x288, x289, x290, x291, x292,
x293, x847, x917, x919, x921,
x929, x931, x933, x935, x937,
x939, x941, x943, x945, x947,
x949, x951, x953, x955, x957,

File Key: a=ltmdirchk.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=ltfssaxes.dtx, q=ltfssstrc.dtx, r=ltfsscnp.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

- 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
 \tc@error x131, x827, x848
 \tc@errorwarn
 ... x21, x97, x99, x736, x737,
 x738, x739, x786, x787, x788, x821
 \tc@fake@euro . x118, x286, x835, x916
 \tc@forcedfalse x780
 \tc@forcedtrue x785
 \tc@oldstylesubst x16, x20
 \tc@subst x98, x130, x820, x847
 \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
 1898, 11104, x226, x631, x880
 \textascendercompwordmark
 1848, x174, x595, x863
 \textasciicute
 1948, 11065, x227, x632, x904
 \textasciibreve
 1896, 11103, x228, x633, x877
 \textasciicaron
 1897, 11102, x229, x634, x878
 \textasciicircum ... 1286, 1555, 11039
 \textasciidieresis
 1936, 11052, x230, x635, x894
 \textasciigrave
 1887, 11031, x231, x636, x875
 \textasciimacron
 1943, 11060, x232, x637, x899
 \textasciitilde 1287, 1556, 11044
 \textasteriskcentered
 1267, 1711, 1858, 11157,
 m130, m136, x140, x503, x558, x870
 \textbackslash . 1268, 1557, 1712, 11038
 \textbaht 1922,
 11107, x235, x641, x1010, x1011
 \textbar 1269, 1558, 1713, 11042
 \textbardbl
 . 1270, 1714, 1902, 11112, m135,
 x147, x293, x504, x566, x707, x883
 \textbf w19
 \textbigcircle 1723,
 1875, 11174, x236, x642, x962, x963
 \textblank
 1855, 11171, x282, x695, x932, x933
 \textborn
 1888, x237, x314, x643, x968, x969
 \textbraceleft
 1271, 1308, 1559, 1715, 11041
 \textbraceright
 1272, 1309, 1560, 1716, 11043
 \textbrokenbar
 1934, 11050, x148, x567, x892
 \textbullet 1273, 1717,
 1904, 11121, x141, x505, x559, x885
 \textcapitalcompwordmark
 1847, x173, x594, x862
 \textcelsius 1905,
 11140, x149, x287, x568, x701, x886
 \textcent 1930, 11046, x150, x569, x889
 \textcentoldstyle 1907,
 x238, x319, x322, x644, x984, x985
 \textcircled . 1279, 1283, 1300, 1301,
 1724, 1876, x175, x176, x570,
 x583, x597, x767, x1032, x1034
 \textcircledP 1941,
 11142, x239, x645, x1026, x1027
 \textcolonmonetary
 1909, 11133, x266, x675, x986, x987
 \textcommaabove 1353, 1355,
 1369, 1370, 1455, 1456, 1694, 1695
 \textcommabelow 1324, 1326,
 1332, 1333, 1697, 1698, 1699, 1700,
 1701, 1702, 1703, 1704, 1705, 1706,
 11194, 11391, 11392, 11393, 11394
 \textcompsubstdefault
 x34, x39, x53, x112, x825
 \textcompwordmark 1290, 1291, 1561, 11109
 \textcopleft 1939,
 x240, x295, x646, x1024, x1025
 \textcopyright 1283,
 1317, 1937, 11053, x151, x570, x895
 \textcurrency 1932, 11048,
 x277, x688, x762, x766, x920, x921
 \textdagger
 .. 1275, 1312, 1719, 1900, 11119,
 m131, m137, x143, x507, x561, x881
 \textdaggerdbl
 .. 1274, 1313, 1718, 1901, 11120,
 m132, m138, x142, x506, x560, x882
 \textdblhyphen
 1859, x242, x296, x648, x934, x935
 \textdblhyphenchar
 1895, x241, x297, x647, x980, x981
 \textdegree 1944, 11061, x152, x571, x900

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\textdied` 1890, x243, x315, x649, x972, x973
- `\textdiscount` 1924, 11132, x244, x650, x1014, x1015
- `\textdiv` . 1961, 11086, x153, x572, x914
- `\textdivorced` 1889, 11177, x245, x651, x970, x971
- `\textdollar` .. 1255, 1307, 1435, 1562, 1792, 1856, 11035, x134, x135, x539, x541, x868, x1040, x1042
- `\textdollaroldstyle` 1906, x246, x320, x321, x652, x982, x983
- `\textdong` 1918, 11137, x267, x676, x1004, x1005
- `\textdownarrow` 1886, 11152, x268, x677, x966, x967
- `\texteightoldstyle` 1869, x216, x311, x620, x952, x953
- `\textellipsis` 1296, 1321, 11122
- `\textemdash` 1256, 1407, 1563, 1779, 11111
- `\textendash` 1257, 1408, 1564, 1780, 11110
- `\textestimated` 1925, 11148, x274, x684, x765, x918, x919
- `\texteuro` 1959, 11138, x285, x286, x699, x763, x915, x916
- `\textexclamdown` 1258, 1409, 1411, 1565, 1781, 11045
- `\textfiveoldstyle` 1866, x217, x308, x621, x946, x947
- `\textfloatsep` M728, M741, M2104, M2154, M2303
- `\textflorin` 1908, 11100, x276, x687, x887
- `\textfont` q291, B205
- `\textfont@name` q285, q291
- `\textfouroldstyle` 1865, x218, x307, x622, x944, x945
- `\textfraction` M1917, M1920, M1944, M1947, M2096, M2297
- `\textfractionsolidus` 1860, 11129, x278, x690, x871
- `\textgravedbl` 1899, 11105, x233, x638, x879
- `\textgreater` ... 1281, 1566, 1734, 11037
- `\textguarani` 1912, x247, x318, x653, x992, x993
- `\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, x284, x697, x1000, x1001
- `\textinterrobangdown` 1917, 11178, x283, x696, x1002, x1003
- `\textit` w21
- `\textlangle` 1871, 11169, x264, x672, x956, x957
- `\textlbrackdbl` 1883, x214, x316, x617, x873
- `\textleaf` 1891, x248, x300, x654, x974, x975
- `\textleftarrow` 1853, 11149, x269, x678, x928, x929
- `\textlegacyasteriskcentered` x515, x710
- `\textlegacybardbl` x515, x711
- `\textlegacybullet` x515, x712
- `\textlegacydagger` x515, x714
- `\textlegacydaggerdbl` x515, x713
- `\textlegacyparagraph` x515, x715
- `\textlegacyperiodcentered` x515, x716
- `\textlegacysection` x515, x717
- `\textless` 1280, 1569, 1733, 11036
- `\textlira` 1914, 11134, x270, x679, x996, x997
- `\textlnot` 1940, 11058, x154, x573, x897
- `\textlquill` 1928, 11130, x249, x655, x1020, x1021
- `\textmarried` 1892, 11176, x250, x656, x976, x977
- `\textmd` w19
- `\textmho` 1874, 11147, x251, x657, x960, x961
- `\textminus` 1872, 11153, x281, x693, x872
- `\textmu` .. 1949, 11066, x280, x692, x905
- `\textmusicalnote` 1893, 11175, x252, x658, x978, x979
- `\textnaira` 1911, 11135, x253, x659, x990, x991
- `\textnineoldstyle` 1870, x219, x312, x623, x954, x955
- `\textnormal` w15
- `\textnumero` 1923, 11141, x275, x685, x1012, x1013
- `\textogonekcentered` .. 1484, 1692, 1693
- `\textohm` 1882, 11146, x279, x691, x764, x917
- `\textonehalf` 1957, 11075, x155, x574, x911
- `\textoneoldstyle` 1862, x220, x304, x624, x938, x939
- `\textonequarter` 1956, 11074, x156, x575, x910
- `\textonesuperior` 1953, 11069, x157, x288, x576, x702, x908

File Key: a=ltmdirchk.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=ltfssaxes.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=ltthyphen.dtx, P=ltluatex.dtx, Q=ltfinal.dtx

- `\textopenbullet` 1926,
11173, x254, x660, x1016, x1017
- `\textordfeminine`
1305, 1938, 11054, x158, x577, x896
- `\textordmasculine`
1306, 1954, 11070, x159, x578, x909
- `\TextOrMath` m130,
m131, m132, m133, m134,
m135, m136, m137, m138, m150
- `\textparagraph` 1276, 1310, 1720, 1950,
11067, m134, x144, x508, x562, x906
- `\textperiodcentered` ... 1277, 1721,
1951, 11068, x145, x509, x563, x907
- `\textpertenthousand`
..... 1489, 1920, 11124, x262,
x263, x669, x1006, x1007, x1044
- `\textperthousand` 1487, 1903, 11123,
x138, x139, x555, x884, x1043
- `\textpeso`
1913, 11139, x255, x661, x994, x995
- `\textpilcrow` 1921,
x256, x313, x662, x1008, x1009
- `\textpm` .. 1945, 11062, x160, x579, x901
- `\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, x161, x580, x869
- `\textquotestraightbase`
.... 1849, x162, x298, x581, x864
- `\textquotestraightdblbase`
.... 1850, x163, x299, x582, x865
- `\texttriangle`
1873, 11170, x265, x673, x958, x959
- `\texttrbrackdbl`
.... 1884, x215, x317, x618, x874
- `\textrecipe`
1915, 11143, x257, x663, x998, x999
- `\textreferencemark` 1952,
11127, x258, x664, x1028, x1029
- `\textregistered` 1300,
1301, 1942, 11059, x164, x583, x898
- `\textrightarrow`
1854, 11151, x271, x680, x930, x931
- `\textrm` w15
- `\textrquill` 1929,
11131, x259, x665, x1022, x1023
- `\textsc` w21
- `\textsection` 1278,
1311, 1576, 1722, 1935, 11051,
m133, x146, x510, x511, x564, x893
- `\textservicemark` 1927,
11144, x260, x666, x1018, x1019
- `\textsevenoldstyle`
1868, x221, x310, x625, x950, x951
- `\textsf` w15
- `\textsixoldstyle`
1867, x222, x309, x626, x948, x949
- `\textsl` w21
- `\textssc` p454, w25, w31, w39
- `\textsterling` 1266, 1319, 1442, 1577,
1799, 1931, 11047, x136, x137,
x540, x548, x890, x1039, x1041
- `\textstyle` j15, u463, B63
- `\textsubscript` I382, I393, I394
- `\textsuperscript` 1303,
1305, 1306, x577, x578, x589, I378
- `\textsurd` 1955,
11168, x261, x667, x1030, x1031
- `\textsw` p450, w25, w30, w38
- `\TextSymbolUnavailable` I3, 1752
- `\textthreeoldstyle`
1864, x223, x306, x627, x942, x943
- `\textthreequarters`
.... 1958, 11076, x166, x586, x912
- `\textthreequartersemdash` ... 1852,
x165, x289, x302, x585, x703, x867
- `\textthreesuperior` 1947,
11064, x167, x290, x587, x704, x903
- `\texttildelow`
.... 1894, 11106, x234, x639, x876
- `\texttimes` 1960, 11080, x168, x588, x913
- `\texttrademark`
1303, 1919, 11145, x169, x589, x888
- `\texttt` w15
- `\texttwelveudash` 1851,
x170, x291, x301, x590, x705, x866
- `\texttwooldstyle`
1863, x224, x305, x628, x940, x941
- `\texttwosuperior` 1946,
11063, x171, x292, x591, x706, x902
- `\textulc` p446, w25, w28, w29, w35, w37
- `\textunderscore` 1288, 1315, 1578, 11040
- `\textup` w21
- `\textuparrow`
1885, 11150, x272, x681, x964, x965
- `\textvisiblespace` .. 1292, 1579, 11172
- `\textwidth` k22,
k79, D301, M79, M144, M201,

File Key: a=ltmdirchk.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=ltfssaxes.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

- M218, M630, M640, M689,
M699, M2223, M2255, Q90, I266
- `\textwon`
1910, I1136, x273, x682, x988, x989
- `\textyen` . 1933, I1049, x172, x592, x891
- `\textzerooldstyle`
l861, x225, x303, x629, x936, x937
- `\tf@size` . o223, o243, o598, q282, q284
- `\TH` l532, I1082, Q481
- `\th` l580, I1088, Q481
- `\thanks` 438, H10, H26
- thebibliography (environment) . . . 471
- `\theequation` . B306, B318, B377, B438
- `\thefootnote` I372, I424, I429, I449
- `\thempfn`
D303, I400, I405, I440, I445, I448
- `\thempfootnote` D303, I374
- `\thepage` k137,
y6, z14, z34, H164, H171, J15,
J32, K23, L14, M244, M275, M1827
- `\Theta` u293
- `\theta` u269
- `\thicklines` F37
- `\thickmuskip` u640, B203
- `\thinlines` F37, F434, F451
- `\thinmuskip` . . . i336, u638, 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` u513
- `\time` a179, a183
- `\times` u390
- `\title` . . . 438, H6, H7, H21, H23, H31
- `\to` u434, u436
- `\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, o148,
o152, o154, o157, o221, o226,
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, s971,
s973, s975, s978, s980, s983,
s986, s1018, s1019, M2209,
M2210, M2211, M2212, N210,
N211, N213, N214, N506, N507
- `\toksdef` b46, b63, b95, P217
- `\tokszero` P217
- `\tolerance`
. b304, o618, o663, o678, L58, L66
- `\top` u314
- `\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, o627
- `\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` u316
- `\triangleleft` u352, u476
- `\triangleright` u353, u476
- `\trivlist` A172, A202, A204, A218,
A240, B428, C89, E66, G35, G37
- `\try@load@fontshape` . . o343, o351,
o397, o502, p423, q474, s208, s225
- `\try@simple@size` q310, q435

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\try@simples` [q393](#), [q399](#), [q403](#)
`\try@size@range` [q101](#), [q310](#), [q386](#)
`\try@size@substitution` .. [q103](#), [q390](#)
`\tryif@simple` [q401](#), [q402](#)
`\trys@simple` [q402](#)
`\ttdef@ult` [t72](#), [t82](#), [t92](#), [t191](#), [x31](#), [x109](#)
`\ttdefault` [t12](#), [t72](#), [t109](#), [t186](#), [t227](#), [u50](#)
`\ttfamily` [t10](#), [t11](#),
[t102](#), [t191](#), [t225](#), [t226](#), [w17](#), [A262](#)
`\ttsubstdefault` .. [u20](#), [u32](#), [x32](#), [x110](#)
`\tw@` [b16](#)
`\two@digits` [a86](#), [a185](#),
[a186](#), [d2](#), [q466](#), [N559](#), [N656](#), [N746](#)
`\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](#), [o337](#), [t116](#),
[t262](#), [t389](#), [t399](#), [t409](#), [u9](#), [u119](#),
[J8](#), [J25](#), [M1898](#), [N158](#), [N173](#),
[N815](#), [N1014](#), [N1017](#), [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](#), [x181](#), [x198](#)
`\uccode` [Q173](#), [Q181](#),
[Q188](#), [Q190](#), [Q193](#), [Q195](#), [Q436](#),
[Q444](#), [Q451](#), [Q453](#), [Q456](#), [Q458](#)
`\uchyph` [b321](#)
`\ulcdefault` [p446](#), [p593](#)
`\ulcshape`
[p446](#), [p560](#), [p561](#), [p592](#), [t248](#), [w29](#)
`\Umathchar` .. [u15](#), [u50](#), [u65](#), [P28](#), [Q247](#)
`\Umathcode`
[b127](#), [x179](#), [x294](#), [A268](#), [Q93](#), [Q430](#)
`\unboldmath` [t314](#)
`\UndeclareTextCommand` [l191](#), [x135](#),
[x137](#), [x139](#), [x263](#), [x511](#), [x1039](#),
[x1040](#), [x1041](#), [x1042](#), [x1043](#), [x1044](#)
`\undefined` . [a9](#), [a11](#), [a17](#), [a57](#), [m92](#),
[m93](#), [m94](#), [m95](#), [m96](#), [m97](#),
[m98](#), [m99](#), [Q101](#), [Q113](#), [Q114](#), [Q134](#)
`\undefinedpagestyle` [L4](#), [L8](#)
`\underbar` [b411](#), [d439](#), [d460](#)
`\underbrace` [u534](#)
- `\underline` ... [377](#), [b411](#), [D358](#), [D359](#)
`\unexpanded` .. [A102](#), [A119](#), [N843](#), [N845](#)
`\unhcopy` [b413](#), [E321](#), [F418](#)
`\unicodedataline`
.. [P143](#), [P146](#), [P160](#), [P161](#), [P162](#)
`\UnicodeEncodingName` .. [l964](#), [l970](#),
[l998](#), [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](#), [x303](#), [x304](#),
[x305](#), [x306](#), [x307](#), [x308](#), [x309](#),
[x310](#), [x311](#), [x312](#), [x313](#), [x314](#),
[x315](#), [x316](#), [x317](#), [x318](#), [x319](#), [x320](#)
`\UnicodeFontFile` [l996](#)
`\UnicodeFontName` [l997](#)
`\UnicodeFontTeXLigatures` .. [l976](#), [l993](#)
`\unicoderead` [P143](#),
[P157](#), [P158](#), [P159](#), [P160](#), [P165](#)
`\uninstall` [P824](#)
`\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](#)

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\unkern` o646
`\unless` P151, P159, P161
`\unlhd` t377
`\unpenalty` o649, o653, w116, A234, A256
`\unquote@name` k230, k236, k275
`\unrestored@protected@xdef`
 . d228, L25, L49, I405, I429, I445
`\unrhd` t379
`\unsetattribute` 586, P82, P227
`\unvcopy` B180
`\Uparrow` u578
`\uparrow` u572
`\upbracefill` u537, u554
`\update@series@target@value`
 t130, t150, t234
`\update@ucllc@with@cyrillic`
 I1441, I1469, I1499, I1507
`\updefault` p550,
 t21, u94, u101, u102, u107, u109
`\Updownarrow` u582
`\updownarrow` u576
`\uplus` u372
`\upper@bound` .. q337, q338, q339, q352
`\uppercase` Q468
`\upshape` 1439, 1726,
 1796, 1878, p548, p549, t19, t20,
 t248, t259, t292, t333, w24, x545
`\Upsilon` u298
`\upsilon` u280
`\use@mathgroup`
 .. o566, o584, o586, q253, s63,
 s92, s424, s526, s529, s996, s1020
`\usebox` D111
`\usecounter` C225, C238
`\usefont` I1506,
 o80, o282, o623, t354, x7, x48, A272
`\UseLegacyTextSymbols` ... x502, x709
`\usepackage` N302, N332, N830
`\UseRawInputEncoding` Q277, Q333, Q380
`\UseTextAccent` ... 1149, 1150, 1188,
 x131, x132, x133, x176, x178,
 x848, x1033, x1034, x1036, x1037
`\UseTextSymbol`
 1150, 1186, x130, x286, x847, x916
`\UTF@four@octets@noexpand` Q467, Q476
`\UTF@three@octets@noexpand`
 Q466, Q475
`\UTF@two@octets@noexpand` Q465, Q474
`\UTFviii@four@octets` Q306, Q311, Q317
`\UTFviii@four@octets@` . Q306, Q317
`\UTFviii@four@octets@combine` . Q341
`\UTFviii@four@octets@noexpand` Q347
`\UTFviii@four@octets@string` .. Q344
`\UTFviii@invalid` Q245, Q338
`\UTFviii@invalid@err` Q303, Q308, Q314
`\UTFviii@invalid@err@` . Q303, Q314
`\UTFviii@three@octets`
 Q305, Q310, Q316
`\UTFviii@three@octets@` Q305, Q316
`\UTFviii@three@octets@combine` Q340
`\UTFviii@three@octets@noexpand` Q346
`\UTFviii@three@octets@string` . Q343
`\UTFviii@two@octets` Q304, Q309, Q315
`\UTFviii@two@octets@` .. Q304, Q315
`\UTFviii@two@octets@combine` .. Q339
`\UTFviii@two@octets@noexpand` . Q345
`\UTFviii@two@octets@string` ... Q342
`\UTFviii@undefined@err`
 Q302, Q307, Q313
`\UTFviii@undefined@err@` Q302, Q313
- V**
- `\v` . 1238, 1387, 1468, 1586, 1587, 1588,
 1592, 1594, 1597, 1599, 1601, 1607,
 1613, 1614, 1615, 1619, 1621, 1624,
 1626, 1628, 1634, 1749, 11189,
 11269, 11270, 11271, 11272, 11281,
 11282, 11315, 11316, 11321, 11322,
 11333, 11334, 11341, 11342, 11345,
 11346, 11368, 11369, 11370, 11371,
 11372, 11373, 11374, 11375, 11376,
 11377, 11378, 11381, 11382, 11383,
 11384, 11387, 11388, x182, x199
`\v@false` B82
`\v@true` B81, B83
`\vadjust` i37,
 i59, i71, i83, i287, i303, I201, I223
`\valign` x122, x839
`\value` 146, m14, K9
`\varbigtriangledown` u356, u359
`\varbigtriangleup` u357, u358
`\varepsilon` j15, u285
`\varphi` u290
`\varpi` u287
`\varrho` u288
`\varsigma` u289
`\vartheta` u286
`\vbadness` b306, M2206
`\vdash` u398
`\vdots` u504
`\vec` u518
`\vector` g219, F113, F435, F452
`\vee` u361, u363
`\verb` A288, A314, A327, A329
`\verb@balance@group` A287,
 A289, A313, A315, A321, A322
`\verb@egroup` A287,
 A290, A313, A316, A322, A326

File Key: a=ltmdirchk.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=ltfssaxes.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

- `\verb@eol@error` ... [A323](#), [A335](#), [A345](#)
`\verbatim` [A260](#)
`\verbatim* (environment)` [A294](#)
`\verbatim@font`
 ... [A233](#), [A254](#), [A262](#), [A336](#), [A346](#)
`\verbatim@nolig@list` [A351](#), [A357](#)
`\verbvisiblespace` [A266](#),
 [A268](#), [A275](#), [A279](#), [A301](#), [A306](#)
`\version@elt` [s18](#), [s31](#), [s32](#), [s256](#), [s257](#),
 [s306](#), [s326](#), [s417](#), [s455](#), [s547](#), [s976](#)
`\version@list` [s16](#),
 [s21](#), [s32](#), [s249](#), [s257](#), [s311](#), [s332](#),
 [s351](#), [s422](#), [s467](#), [s497](#), [s552](#), [s989](#)
`\Vert` [u565](#), [u567](#)
`\vert` [u570](#)
`\vfil` [b400](#), [x123](#), [x126](#),
 [x840](#), [x843](#), [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](#), [u552](#), [u553](#),
 [u555](#), [u556](#), [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` [u360](#), [u362](#)
`\whatsit` [P185](#)
`\widehat` [u521](#)
`\widetilde` [u520](#)
`\widowpenalties` [b106](#)
`\widowpenalty` [b313](#), [o636](#)
`\width` [D30](#)
`\wlog` [a100](#), [b40](#),
 [b145](#), [b230](#), [b243](#), [b273](#), [b288](#),
 [N175](#), [P6](#), [P7](#), [P8](#), [P52](#), [Q46](#), [Q539](#)
`\wp` [u306](#)
`\wr` [u376](#)
`\wrong@fontshape` [o347](#), [o484](#)
- X**
- `\x` [o304](#), [o305](#)
`\x@protect` [d206](#), [d217](#), [d265](#), [d292](#)
`\xe@alloc@` [Q42](#), [Q52](#)
`\xe@alloc@intercharclass` [Q21](#)
`\xe@ch@ck` [Q43](#), [Q47](#)
`\XeTeXcharclass` [o611](#), [Q25](#),
 [Q33](#), [Q40](#), [Q53](#), [Q59](#), [Q68](#), [Q75](#)
`\XeTeXcharclassCL` [Q106](#)
`\XeTeXcharclassCM` [Q110](#)
`\XeTeXcharclassEX` [Q107](#)
`\XeTeXcharclassID` [Q104](#)
`\XeTeXcharclassIS` [Q108](#)
`\XeTeXcharclassNS` [Q109](#)
`\XeTeXcharclassOP` [Q105](#)
`\XeTeXcharglyph` [1994](#)
`\XeTeXdashbreakstate` [Q204](#)
`\XeTeXglyph` [1994](#)
`\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` [u295](#)
`\xi` [u275](#)
`\xtxHanGlue` [Q113](#),
 [Q137](#), [Q145](#), [Q146](#), [Q147](#), [Q148](#),
 [Q149](#), [Q150](#), [Q151](#), [Q152](#), [Q153](#)
`\xtxHanSpace` .. [Q114](#), [Q138](#), [Q139](#),
 [Q140](#), [Q141](#), [Q142](#), [Q143](#), [Q144](#)
- Y**
- `\year` [a185](#), [c14](#), [N656](#), [N746](#)
`\yxdim` [F354](#)
- Z**
- `\Z` [Q185](#), [Q427](#), [Q448](#)
`\z` [Q176](#), [Q428](#), [Q439](#)
`\z@` [b298](#)
`\z@skip` [b298](#)
`\zap@space` [k152](#), [k169](#), [t245](#),
 [K29](#), [N108](#), [N199](#), [N281](#), [N365](#),
 [N386](#), [N403](#), [N414](#), [N431](#), [N578](#)
`\zeta` [u267](#)

File Key: a=ltmdirchk.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=ltfssaxes.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