

The L^AT_EX 2_ε Sources

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

2016/03/31 Patch level

This file is maintained by the L^AT_EX Project team.
Bug reports can be opened (category `latex`) at
<http://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)	6
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 (???)	7
3.8	MACINTOSH (OzTeX 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	lplain.dtx	14
9	Plain T_EX	14
c	ltxvers.dtx	32
10	Version Identification	32
d	ltxdefns.dtx	34
11	Definitions	34
	11.1 Initex initialisations	34
	11.2 Saved versions of T _E X primitives	34
	11.3 Command definitions	35
	11.4 Robust commands and protect	43
	11.5 Internal defining commands	46
e	ltxalloc.dtx	49
12	Counters	49
f	ltxcntrl.dtx	51
13	Program control structure	51
g	ltxerror.dtx	55
14	Error handling	55
	14.1 General commands	55
	14.2 Specific errors	60
h	ltxpar.dtx	64
15	Paragraphs	64
	15.1 Implementation	64
i	ltxspace.dtx	66

16 Spacing	66
16.1 User Commands	66
16.2 Chris' comments	66
16.3 Some immediate actions	68
16.4 The code	69
16.5 Vertical spacing	74
16.6 Horizontal space (and breaks)	77
j Itlogos.dtx	80
17 Logos	80
k Itfiles.dtx	81
18 File Handling	81
18.1 Safe Input Macros	87
18.2 Listing files	89
l Itoutenc.dtx	91
19 Font encodings	91
19.1 Removing encoding-specific commands	93
19.2 The order of declarations	94
19.3 Docstrip modules	94
19.4 Definitions for the kernel	94
19.4.1 Declaration commands	94
19.4.2 Hyphenation	101
19.4.3 Miscellania	102
19.4.4 Default encodings	102
19.4.5 Math material	104
19.5 Definitions for the OT1 encoding	105
19.6 Definitions for the T1 encoding	107
19.7 Definitions for the OMS encoding	112
19.8 Definitions for the OML encoding	112
19.9 Definitions for the OT4 encoding	113
19.10 Definitions for the TS1 encoding	115
20 Package files	119
20.1 The fontenc package	119
20.2 The textcomp package	121
20.2.1 Supporting oldstyle digits	129
20.2.2 Subset encoding defaults	130
m Itcounts.dtx	132
21 Counters and Lengths	132
21.1 Environment Counter Macros	132

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

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

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

54 List, and related environments	269
54.1 List and Trivlist	270
54.2 Vertical Spacing (skips)	271
54.3 Penalties	271
54.4 Horizontal Spacing (dimens)	271
54.5 Default Values	272
54.6 Itemize and Enumerate	282
B Itboxes.dtx	284
55 L^AT_EX Box commands	284
55.1 Some low-level constructs	295
C Ittab.dtx	296
56 Tabbing, Tabular and Array Environments	296
56.1 tabbing	296
56.2 array and tabular environments	304
D Itpictur.dtx	318
57 Picture Mode	318
57.1 Curves	337
E Itthm.dtx	340
58 Theorem Environments	340
F Itsect.dtx	344
59 Sectioning Commands	344
59.1 The Title	344
59.2 Sectioning	345
59.2.1 Initializations	351
59.3 Table of Contents etc.	351
59.3.1 Convention	351
59.3.2 Commands	351
G Itfloat.dtx	354
60 Floats	354
60.1 Floating Environments	354
60.2 Footnotes	367

H	ltxglo.dtx	374
61	Index and Glossary Generation	374
I	ltxbibl.dtx	376
62	Bibliography Generation	376
	62.1 Default definitions	379
J	ltxpage.dtx	380
63	Page styles and related commands	380
	63.1 Page Style Commands	380
	63.2 How a page style makes running heads and feet	380
	63.3 marking conventions	380
K	ltxoutput.dtx	383
64	Output Routine	383
	64.1 Floats	383
	64.1.1 Kludgeins	436
	64.1.2 Float control	437
	64.1.3 Float placement parameters	449
L	ltxclass.dtx	452
65	Introduction	452
66	User interface	452
	66.1 Option processing	453
67	Class and Package interface	453
	67.1 Class name and version	453
	67.2 Package name and version	454
	67.3 Requiring other packages	454
	67.4 Declaring new options	455
	67.5 Safe Input Macros	455
68	Implementation	456
	68.1 Hooks	466
	68.2 Providing shipment	467
69	After Preamble	469
M	ltxhyphen.dtx	471

N	lualatex.dtx	473
70	Overview	473
71	Core T_EX functionality	473
72	Plain T_EX interface	474
73	Lua functionality	474
73.1	Allocators in Lua	474
73.2	Lua access to T _E X register numbers	474
73.3	Module utilities	476
73.4	Callback management	476
74	Implementation	477
74.1	Minimum LuaT _E X version	477
74.2	Older L ^A T _E X/Plain T _E X setup	477
74.2.1	Fixes to <code>etex.src/etex.sty</code>	477
74.2.2	luatex specific settings	478
74.3	Attributes	479
74.4	Category code tables	479
74.5	Named Lua functions	481
74.6	Custom whatsits	481
74.7	Lua bytecode registers	481
74.8	Lua chunk registers	482
74.9	Lua loader	482
74.10	Lua module preliminaries	483
74.11	Lua module utilities	483
74.11.1	Module tracking	483
74.11.2	Module messages	484
74.12	Accessing register numbers from Lua	485
74.13	Attribute allocation	486
74.14	Custom whatsit allocation	487
74.15	Bytecode register allocation	487
74.16	Lua chunk name allocation	487
74.17	Lua callback management	488
74.17.1	Housekeeping	488
74.17.2	Handlers	490
74.17.3	Public functions for callback management	491
O	lfinal.dtx	496
75	Final settings	496
75.1	Debugging	496
75.2	Typesetting parameters	496
75.3	Lccodes for hyphenation	498
75.4	Hyphenation	501
75.5	Font loading	501
75.6	Input encoding	502
75.7	Lccodes and uccodes	503

75.8 Applying Patch files	505
75.9 Freeing Memory	505
75.10 Initialise file list	506
75.11 Dumping the format	506

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'\f=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

```

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

```

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

```

2.2 Some bits of 2e

```

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

```

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

```

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

```

```

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

```

`\strip@prefix`

```

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

```

3 texsys.cfg

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

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

```

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

```

START

3.1 texsys.cfg

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

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

The macros that must be defined are:

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

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

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

`<dir><filename><space>`

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

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

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

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

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

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

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

3.2 UNIX (web2c)

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

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



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

3.3 UNIX (other)

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

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

3.4 MSDOS (emtex)

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

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

3.5 MSDOS (other)

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

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

3.6 VMS (DECUS T_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.

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

3.7 VMS (???)

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

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

```

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

```

3.8 MACINTOSH (OzTeX 1.6)

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

```

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

```

3.9 MACINTOSH (other)

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

```

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

```

3.10 FAKE EXAMPLE

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

the following code would possibly work:

```

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

```

END

```

165 \immediate\closeout15

```

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

```

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

```

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

```
172 <dircheck>\input texsys.cfg
```

4 Setting \@currdir

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

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

`\today` The current date and time stamp.

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

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

```
182 \immediate \openout 15 = texsys.aux
183 \immediate \write 15 { \today ^^ J }
184 \immediate \closeout 15 %
```

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

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

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

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

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

```

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

```

Otherwise `\@currdir` was defined in `texsys.cfg`. In this case check that the syntax specified works on this system. (In case a complete L^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.

```

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

```

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

```

218 \fi

219 \immediate\closeout15 %
220 \endgroup

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

```

Stop here if the file is being used unstripped.

```

225 (*docstrip)
226 \relax\endinput
227 </docstrip)

```

5 Setting `\input@path`

Earlier versions of this file attempted to automatically test whether `\input@path` was required, and interactively prompt for a path if necessary. This was not found to be very reliable. The first-time installer of L^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.

```

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

6 Filename Parsing

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

```

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

```

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

Search for the last /.

```

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

```

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

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

```

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

Search for the last].

```

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

```

262     \fi
263     \reserved@a}

264 \else\def\reserved@a{:}\ifx\@currdir\reserved@a
\filename@parse was not specified in texsys.cfg, but \@currdir looks like Mac-
intosh...
265     \typeout{^^JDefining Mac style filename parser.^^J}
266     \def\filename@parse#1{%
267         \let\filename@area\@empty
268         \expandafter\filename@path#1:\}

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

277 \else

\filename@parse was not specified in texsys.cfg. So just make a simple parser
that always sets \filename@area to empty.
278     \typeout{^^JDefining generic filename parser.^^J}
279     \def\filename@parse#1{%
280         \let\filename@area\@empty
281         \expandafter\filename@simple#1.\}
282 \fi\fi\fi

    \filename@simple is used by all three versions. Finally we can split off the
extension.
283 \def\filename@simple#1.#2\{%
284     \ifx\#2\%
285         \let\filename@ext\relax
286     \else
287         \edef\filename@ext{\filename@dot#2\}%
288     \fi
289     \edef\filename@base{#1}}

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

291 \else

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

```

7 T_EX Versions

`\@TeXversion` T_EX versions older than than 3.141 require `\@TeXversion` to be set. This can be determined automatically due to a trick suggested by Bernd Raichle. (Actually this will not always get the correct version number, eg T_EX3.14 would be detected as T_EX3, but L^AT_EX only needs to take account of T_EX's older than 3, or between 3 and 3.14.

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

307 </dircheck>
```

8 ltxcheck.tex

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

File b

ltpplain.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 \outer\def^^L{\par}% ascii form-feed is \outer\par
12 \message{catcodes,}
```

We had to define the `\catcodes` right away, before the message line, since `\message` uses the `{` and `}` characters. When INITEX (the T_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.

The following counters are reserved:

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

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

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

1, 3, 5, 7, 9,

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

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

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

```

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

```

```

\insc@unt The insertion counter and most recent allocation.
\allocationnumber 37 \countdef\insc@unt=20
38 \countdef\allocationnumber=21

\m@ne The constant -1.
39 \countdef\m@ne=22 \m@ne=-1

\wlog Write on log file (only)
40 \def\wlog{\immediate\write\m@ne}

\count@ Here are abbreviations for the names of scratch registers that don't need to be
\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.
\newread
\newwrite
\newlanguage File b: ltplain.dtx Date: 2015/11/18 Version v2.2b

```

L^AT_EX change: remove `\outer` from `\newcount` and `\newdimen` (FMi) This is necessary to use `\newcount` inside `\if...` later on. Also remove from `\newskip`, `\newbox` `\newwrite` and `\newfam` (DPC) to save later redefinition.

```

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

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

```

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

```

56 \def\newbox   {\e@alloc\box
57               {\ifnum\allocationnumber<\@ccclvi
58                 \expandafter\chardef
59                 \else
60                 \expandafter\e@alloc@chardef
61                 \fi}
62               {\count14}\insc@unt\float@count}
63 \def\newtoks  {\e@alloc\toks \toksdef{\count15}\m@ne\e@alloc@top}
64 \def\newread  {\e@alloc\read  \chardef{\count16}\m@ne\sixt@@n}

```

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

```

65 \ifx\directlua@undefined
66 \def\newwrite {\e@alloc\write \chardef{\count17}\m@ne\sixt@@n}
67 \else
68 \def\newwrite {\e@alloc\write
69               {\ifnum\allocationnumber=18 \allocationnumber19\fi
70               \global\chardef}%
71               {\count17}%
72               \m@ne
73               {128}}
74 \fi

```

```

75 \def\new@mathgroup
76   {\e@alloc\mathgroup\chardef{\count18}\m@ne\e@mathgroup@top}
77 \def\newlanguage {\e@alloc\language \chardef{\count19}\m@ne@\ccclvi}
78 \let\newfam\new@mathgroup
79 </2ekernel | latexrelease>

```

```

80 (latexrelease)\EndIncludeInRelease
81 (latexrelease)\IncludeInRelease{0000/00/00}%
82 (latexrelease)           {\newcount}{Extended Allocation}%
83 (latexrelease)\def\newcount{\alloc@0\count\countdef\insc@unt}
84 (latexrelease)\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}
85 (latexrelease)\def\newskip{\alloc@2\skip\skipdef\insc@unt}
86 (latexrelease)\def\newmuskip{\alloc@3\muskip\muskipdef\@ccclvi}
87 (latexrelease)\def\newbox{\alloc@4\box\chardef\insc@unt}
88 (latexrelease)\def\newtoks{\alloc@5\toks\toksdef\@ccclvi}
89 (latexrelease)\def\newread{\alloc@6\read\chardef\sixt@@n}
90 (latexrelease)\def\newwrite{\alloc@7\write\chardef\sixt@@n}
91 (latexrelease)\def\new@mathgroup{\alloc@8\fam\chardef\sixt@@n}
92 (latexrelease)\def\newlanguage{\alloc@9\language\chardef\@ccclvi}

```

```

93 (latexrelease)\let\newfam\new@mathgroup
94 (latexrelease)\EndIncludeInRelease

\@alloc@chardef The upper limit of extended registers, which leaves this number (eg \dimen32767)
\@alloc@top always unallocated by these macros. cf traditional \dimen255.
95 (*2kernel | latexrelease)
96 (latexrelease)\IncludeInRelease{2015/01/01}%
97 (latexrelease) {\@alloc@chardef}{Extended Allocation}%

98 \ifx\directlua\@undefined
99 \ifx\widowpenalties\@undefined

classic tex has 28 registers.
100 \mathchardef\@alloc@top=255
101 \let\@alloc@chardef\chardef
102 \else

etex and xetex have 215 registers.
103 \mathchardef\@alloc@top=32767
104 \let\@alloc@chardef\mathchardef
105 \fi
106 \else

luatex has 216 registers.
107 \chardef\@alloc@top=65535
108 \let\@alloc@chardef\chardef
109 \fi

110 (/2kernel | latexrelease)
111 (latexrelease)\EndIncludeInRelease
112 (latexrelease)\IncludeInRelease{0000/00/00}%
113 (latexrelease) {\@alloc@chardef}{Extended Allocation}%
114 (latexrelease)\let\@alloc@top\@undefined
115 (latexrelease)\let\@alloc@chardef\@undefined
116 (latexrelease)\EndIncludeInRelease

\@mathgroup@top The upper limit of extended math groups (\fam) 16 in classic TEX and e-TEX, but
256 in Unicode TeX variants.
117 (*2kernel | latexrelease)
118 (latexrelease)\IncludeInRelease{2015/01/01}%
119 (latexrelease) {\@mathgroup@top}{Extended Allocation}%

120 \ifx\Umathcode\@undefined

classic and e tex have 16 fam (0–15).
121 \chardef\@mathgroup@top=16
122 \else

xetex and luatex have 256 fam (0–255).
123 \chardef\@mathgroup@top=256
124 \fi

125 (/2kernel | latexrelease)
126 (latexrelease)\EndIncludeInRelease
127 (latexrelease)\IncludeInRelease{0000/00/00}%
128 (latexrelease) {\@mathgroup@top}{Extended Allocation}%
129 (latexrelease)\let\@mathgroup@top\@undefined
130 (latexrelease)\EndIncludeInRelease

```

`\e@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
    \e@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.
131 (*2kernel | latexrelease)
132 (latexrelease)\IncludeInRelease{2015/01/01}{\e@alloc}{Extended Allocation}%

133 \def\e@alloc#1#2#3#4#5#6{%
134   \global\advance#3\@ne
135   \e@ch@ck{#3}{#4}{#5}#1%
136   \allocationnumber#3\relax
137   \global#2#6\allocationnumber
138   \wlog{\string#6=\string#1\the\allocationnumber}}%

139 (/2kernel | latexrelease)
140 (latexrelease)\EndIncludeInRelease
141 (latexrelease)\IncludeInRelease{0000/00/00}{\e@alloc}{Extended Allocation}%
142 (latexrelease)\let\e@alloc\@undefined
143 (latexrelease)\EndIncludeInRelease
144 (*2kernel)

```

`\e@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`.

```

145 (/2kernel)
146 (*2kernel | latexrelease)
147 (latexrelease)\IncludeInRelease{2015/10/01}
148 (latexrelease)           {\e@ch@ck}{Extended Allocation (checking)}%

149 \gdef\e@ch@ck#1#2#3#4{%
150   \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).

```

151   \ifnum#1=#2\relax
152     \global#1\@cclvi
153     \ifx\count#4\global\advance#1 10 \fi
154   \fi

```

Check we are below the extended limit.

```

155   \ifnum#1<#3\relax
156   \else
157     \errmessage{No room for a new \string#4}%
158   \fi
159 \fi}%
160 (latexrelease)\EndIncludeInRelease
161 (latexrelease)\IncludeInRelease{2015/01/01}%
162 (latexrelease)           {\e@ch@ck}{Extended Allocation (checking)}%
163 (latexrelease)\gdef\e@ch@ck#1#2#3#4{%
164 (latexrelease)   \ifnum#1<#2\else

```

```

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

179 <latexrelease> \IncludeInRelease{2015/01/01}%
180 <latexrelease> {\extrafloats}{Extra floats}%

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

```

\extrafloats

```

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

```

```

211 </2ekernel | latexrelease>
212 <latexrelease>\EndIncludeInRelease
213 <latexrelease>\IncludeInRelease{0000/00/00}%
214 <latexrelease>          {\extrafloats}{Extra floats}%
215 <latexrelease>\let\float@count\@undefined
216 <latexrelease>\let\extrafloats\@undefined
217 <latexrelease>\EndIncludeInRelease
218 <*2ekernel>

\alloc@

219 \def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
220 \ch@ck#1#4#2% make sure there's still room
221 \allocationnumber\count1#1%
222 \global#3#5\allocationnumber
223 \wlog{\string#5=\string#2\the\allocationnumber}}

\newinsert

224 </2ekernel>
225 <*2ekernel | latexrelease>
226 <latexrelease>\IncludeInRelease{2015/10/01}
227 <latexrelease>          {\newinsert}{Extended \newinsert}%

228 \ifx\numexpr\@undefined

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

229 \def\newinsert#1{\global\advance\insc@unt \m@ne
230 \ch@ck0\insc@unt\count
231 \ch@ck1\insc@unt\dimen
232 \ch@ck2\insc@unt\skip
233 \ch@ck4\insc@unt\box
234 \allocationnumber\insc@unt
235 \global\chardef#1\allocationnumber
236 \wlog{\string#1=\string\insert\the\allocationnumber}}

237 \else

The highest register allowed with \insert.

238 \ifx\directlua\@undefined
239 \chardef\@insert@top255
240 \else
241 \chardef\@insert@top\@alloc@top
242 \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.

243 \def\newinsert#1{%
244 \@tempswafalse
245 \ifnum\count10<\insc@unt
246 \ifnum\count11<\insc@unt
247 \ifnum\count12<\insc@unt
248 \ifnum\count14<\insc@unt
249 \@tempswatru
250 \fi\fi\fi\fi
251 \if@tempswa
252 \global\advance\insc@unt\m@ne
253 \allocationnumber\insc@unt

```

```

254 \else
255   \extrafloats\@ne
256   \@next\@currbox\@freelist
257   {\ifnum\@currbox<\e@insert@top
258     \allocationnumber\@currbox
259     \else
260     \ch@ck0\m@ne\insert
261     \fi}%
262   {\ch@ck0\m@ne\insert}%
263 \fi
264 \global\chardef#1\allocationnumber
265 \wlog{\string#1=\string\insert\the\allocationnumber}%
266 }

267 \fi
268 (/2ekernel | latexrelease)

269 (latexrelease)\EndIncludeInRelease
270 (latexrelease)\IncludeInRelease{0000/00/00}%
271 (latexrelease)           {\newinsert}{Extended \newinsert}%
272 (latexrelease)\let\e@insert@top\@undefined
273 (latexrelease)\def\newinsert#1{\global\advance\insc@unt \m@ne
274 (latexrelease) \ch@ck0\insc@unt\count
275 (latexrelease) \ch@ck1\insc@unt\dimen
276 (latexrelease) \ch@ck2\insc@unt\skip
277 (latexrelease) \ch@ck4\insc@unt\box
278 (latexrelease) \allocationnumber\insc@unt
279 (latexrelease) \global\chardef#1\allocationnumber
280 (latexrelease) \wlog{\string#1=\string\insert\the\allocationnumber}}
281 (latexrelease)\EndIncludeInRelease
282 (*2ekernel)

\ch@ck

283 \gdef\ch@ck#1#2#3{%
284   \ifnum\count1#1<#2\else
285     \errmessage{No room for a new #3}%
286   \fi}

\newhelp

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

\maxdimen Here are some examples of allocation.
\hideskip 288 \newdimen\maxdimen \maxdimen=16383.99999pt % the largest legal <dimen>
289 \newskip\hideskip \hideskip=-1000pt plus 1fill % negative but can grow

\p@
\z@
290 \newdimen\p@ \p@=1pt % this saves macro space and time
\z@skip 291 \newdimen\z@ \z@=0pt % can be used both for 0pt and 0
\voidb@x 292 \newskip\z@skip \z@skip=0pt plus0pt minus0pt
293 \newbox\voidb@x % permanently void box register

294 \message{compatibility for TeX 2, }

```


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

```
295 \ifx\@undefined\inputlineno
296   \newcount\inputlineno
```

This could be used to detect that an old T_EX is in force

```
297   \inputlineno-1
```

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

```
298   \ifx\@undefined\language
299     \newcount\language
300   \fi
301   \newcount\lefthyphenmin
302   \newcount\riquiryphenmin
303   \newcount\errorcontextlines
304   \newcount\holdinginserts
305   \newdimen\emergencystretch
306   \newcount\badness
307   \let\noboundary\relax
308   \newcount\setlanguage
309 \fi
```

Assign initial values to T_EX's parameters

```
310 \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.

```
311 \pretolerance=100
312 \tolerance=200 % INITEX sets this to 10000
313 \hbadness=1000
314 \vbadness=1000
315 \linepenalty=10
316 \hyphenpenalty=50
317 \exhyphenpenalty=50
318 \binoppenalty=700
319 \relpenalty=500
320 \clubpenalty=150
321 \widowpenalty=150
322 \displaywidowpenalty=50
323 \brokenpenalty=100
324 \predisplaypenalty=10000
    \postdisplaypenalty=0
    \interlinepenalty=0
    \floatingpenalty=0, set during \insert
    \outputpenalty=0, set before TeX enters \output
325 \doublehyphendemerits=10000
326 \finalhyphendemerits=5000
327 \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
328 \tracinglostchars=1

\tracingcommands=0
\tracingrestores=0
\language=0
329 \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
330 \defaultthyphenchar='\-
331 \defaultskewchar=-1

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

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

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

336 \hfuzz=0.1pt
337 \vfuzz=0.1pt
338 \overfullrule=5pt
339 \maxdepth=4pt
340 \splitmaxdepth=\maxdimen
341 \boxmaxdepth=\maxdimen

\lineskiplimit=0pt, changed by \normalbaselines
342 \delimitershortfall=5pt
343 \nulldelimiterspace=1.2pt
344 \scriptspace=0.5pt

```

```

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

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

\baselineskip=0pt, changed by \normalbaselines
\lineskip=0pt, changed by \normalbaselines
346 \parskip=0pt plus 1pt
347 \abovedisplayskip=12pt plus 3pt minus 9pt
348 \abovedisplayshortskip=0pt plus 3pt
349 \belowdisplayskip=12pt plus 3pt minus 9pt
350 \belowdisplayshortskip=7pt plus 3pt minus 4pt

\leftskip=0pt
\rightskip=0pt
351 \topskip=10pt
352 \splittopskip=10pt

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

\normalbaselineskip We also define special registers that function like parameters:
\normallineskip 354 \newskip\normalbaselineskip \normalbaselineskip=12pt
\normallineskiplimit 355 \newskip\normallineskip \normallineskip=1pt
356 \newdimen\normallineskiplimit \normallineskiplimit=0pt

\interfootlinepenalty
357 \newcount\interfootnotelinepenalty \interfootnotelinepenalty=100

Definitions for preloaded fonts

\magstephalf
\magstep 358 \def\magstephalf{1095 }
359 \def\magstep#1{\ifcase#1 \@m\or 1200\or 1440\or 1728\or
360 2074\or 2488\fi\relax}

Macros for setting ordinary text

\frenchspacing
\nonfrenchspacing 361 \def\frenchspacing{\sfcode'\.\@m \sfcode'\?\@m \sfcode'\!\@m
362 \sfcode'\:\@m \sfcode'\;\@m \sfcode'\,\@m}
363 \def\nonfrenchspacing{\sfcode'\.3000\sfcode'\?3000\sfcode'\!3000%
364 \sfcode'\:2000\sfcode'\;1500\sfcode'\,1250 }

```

```

\normalbaselines
365 \def\normalbaselines{\lineskip\normallineskip
366 \baselineskip\normalbaselineskip \lineskiplimit\normallineskiplimit}

\M Save a bit of space by using \let here.
\I 367 \def^^M{\ } % control <return> = control <space>
368 \let^^I^^M % same for <tab>

\lq
\rq 369 \def\lq{'
370 \def\rq{'}

\lbrack
\rbrack 371 \def\lbrack{[}
372 \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.
373 \def \aa {\r a}
374 \def \AA {\r A}

\endgraf
\endline 375 \let\endgraf=\par
376 \let\endline=\cr

\space
377 \def\space{ }

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

\null
379 \def\null{\hbox{}}

\bgroup
\egroup 380 \let\bgroup={
381 \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{...
382 {\catcode'\^^M=\active % these lines must end with %
383 \gdef\obeylines{\catcode'\^^M\active \let^^M\par}%
384 \global\let^^M\par} % this is in case ^^M appears in a \write
385 \def\obeyspaces{\catcode'\ \active}
386 {\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).
387 \long\def \loop #1\repeat{%
388 \def\iterate{#1\relax % Extra \relax
389 \expandafter\iterate\fi

```

```

390             }%
391 \iterate
392 \let\iterate\relax
393 }

```

This setting of `\repeat` is needed to make `\loop... \if... \repeat` skippable within another `\if...`

```
394 \let\repeat=\fi
```

L^AT_EX defines `\smallskip`, etc. in `ltspace.dtx`.

```

\nointerlineskip
\offinterlineskip 395 \def\nointerlineskip{\prevdepth-\@m\p@}
396 \def\offinterlineskip{\baselineskip-\@m\p@
397 \lineskip\z@ \lineskiplimit\maxdimen}

```

```

\vglue
\hglue 398 \def\vglue{\afterassignment\vgl@\skip@=}
399 \def\vgl@{\par \dimen@\prevdepth \hrule \@height\z@
400 \nobreak\vskip\skip@ \prevdepth\dimen@}
401 \def\hglue{\afterassignment\hgl@\skip@=}
402 \def\hgl@{\leavevmode \count@\spacefactor \vrule \@width\z@
403 \nobreak\hskip\skip@ \spacefactor\count@}

```

L^AT_EX defines `~` in `ltdefns.dtx`.

```

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

```

```

\break
\nobreak 405 \def\break{\penalty-\@M}
\allowbreak 406 \def\nobreak{\penalty \@M}
407 \def\allowbreak{\penalty \z@}

```

```

\filbreak
\goodbreak 408 \def\filbreak{\par\vfil\penalty-200\vfilneg}
409 \def\goodbreak{\par\penalty-500 }

```

`\eject` Define `\eject` as in plain T_EX but define `\supereject` only in the compatibility file.

```
410 \def\eject{\par\break}
```

```

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

```

```

\smallbreak
\medbreak 412 \def\smallbreak{\par\ifdim\lastskip<\smallskipamount
\bigbreak 413 \removelastskip\penalty-50\smallskip\fi}
414 \def\medbreak{\par\ifdim\lastskip<\medskipamount
415 \removelastskip\penalty-100\medskip\fi}
416 \def\bigbreak{\par\ifdim\lastskip<\bigskipamount
417 \removelastskip\penalty-200\bigskip\fi}

```

```

\m@th
418 \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?).
419 \def\underbar#1{\underline{\sbox\tw@{#1}\dp\tw@\z@\box\tw@}}

\strutbox LATEX sets \strutbox in \set@fontsize.
\strut
420 \newbox\strutbox
421 \def\strut{\relax\ifmode\copy\strutbox\else\unhcopy\strutbox\fi}

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

\narrower
423 \def\narrower{%
424 \advance\leftskip\parindent
425 \advance\rightskip\parindent}

LATEX defines \ae and similar commands elsewhere.
426 \chardef\%= '\%
427 \chardef\&= '\&
428 \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
429 \def\leavevmode{\unhbox\voidb@x}

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

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

\oalign
\o@lign 432 \def\oalign#1{\leavevmode\vtop{\baselineskip\z@skip \lineskip.25ex%
\oalign 433 \ialign{##\crr#1\crr}}
434 \def\o@lign{\lineskiplimit\z@ \oalign}
435 \def\ooalign{\lineskiplimit-\maxdimen \oalign}

\sh@ft The definition of this macro in plain.tex was improved in about 1997; but as a
result its usage was changed and its new definition is not appropriate for LATEX.
Since the version given here has been in use by LATEX 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.
436 \def\sh@ft#1{\dimen@.00#1ex\multiply\dimen@\fontdimen1\font
437 \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`).

```
438 \def\ltx@sh@ft #1{%
439   \dimen@ #1%
440   \kern \strip@pt
441   \fontdimen1\font \dimen@
442 } % 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.

```
443 \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.

```
444 \def\dotfill{%
445   \leavevmode
446   \cleaders \hb@xt@ .44em{\hss.\hss}\hfill
447   \kern\z@}
```

INITEX sets `\sfcode x=1000` for all `x`, except that `\sfcode X=999` for uppercase letters. The following changes are needed:

```
448 \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@
  \size6.5truein\size8.9truein\dimen\footins8truein}
```

`\showoverfull` The following commands are used in debugging:

```
449 \def\showoverfull{\tracingonline\@ne}
```

`\showoutput`

`\loggingoutput`

```
450 \gdef\loggingoutput{\tracingoutput\@ne
451   \showboxbreadth\maxdimen\showboxdepth\maxdimen\errorstopmode}
452 \gdef\showoutput{\loggingoutput\showoverfull}
453 \endkernel)
```

`\tracingall`

`\loggingall`

```
454 (latexrelease)\IncludeInRelease{2015/01/20}{\loggingall}{etex tracing}%
455 (*2ekernel | latexrelease)
```

```

456 \ifx\tracingscantokens\@undefined
457 \gdef\loggingall{%
458   \tracingstats\tw@
459   \tracingpages\@ne
460   \tracinglostchars\@ne
461   \tracingparagraphs\@ne
462   \errorcontextlines\maxdimen
463   \loggingoutput
464   \tracingmacros\tw@
465   \tracingcommands\tw@
466   \tracingrestores\@ne
467 }%
468 \else
469 \gdef\loggingall{%
470   \tracingstats\tw@
471   \tracingpages\@ne
472   \tracinglostchars\tw@
473   \tracingparagraphs\@ne
474   \tracinggroups\@ne
475   \tracingifs\@ne
476   \tracingscantokens\@ne
477   \tracingnesting\@ne
478   \errorcontextlines\maxdimen
479   \loggingoutput
480   \tracingmacros\tw@
481   \tracingcommands\thr@@
482   \tracingrestores\@ne
483   \tracingassigns\@ne
484 }%
485 \fi
486 \gdef\tracingall{\showoverfull\loggingall}
487 </2ekernel | latexrelease>
488 <latexrelease>\EndIncludeInRelease
489 <latexrelease>\IncludeInRelease{0000/00/00}{\loggingall}{etex tracing}%
490 <latexrelease>\gdef\loggingall{\tracingcommands\tw@\tracingstats\tw@
491 <latexrelease> \tracingpages\@ne\tracinglostchars\@ne
492 <latexrelease> \tracingmacros\tw@\tracingparagraphs\@ne\tracingrestores\@ne
493 <latexrelease> \errorcontextlines\maxdimen\loggingoutput}
494 <latexrelease> \gdef\tracingall{\loggingall\showoverfull}
495 <latexrelease>\EndIncludeInRelease

```

\tracingnone

\hideoutput

```

496 <latexrelease>\IncludeInRelease{2015/01/20}{\tracingnone}%
497 <latexrelease> \turn off etex tracing}%
498 <*2ekernel | latexrelease>
499 \ifx\tracingscantokens\@undefined
500 \def\tracingnone{%
501   \tracingonline\z@
502   \tracingcommands\z@
503   \showboxdepth\m@ne
504   \showboxbreadth\m@ne
505   \tracingoutput\z@
506   \errorcontextlines\m@ne
507   \tracingrestores\z@

```



```

508 \tracingparagraphs\z@
509 \tracingmacros\z@
510 \tracinglostchars\@ne
511 \tracingpages\z@
512 \tracingstats\z@
513 }%
514 \else
515 \def\tracingnone{%
516 \tracingassigns\z@
517 \tracingrestores\z@
518 \tracingonline\z@
519 \tracingcommands\z@
520 \showboxdepth\m@ne
521 \showboxbreadth\m@ne
522 \tracingoutput\z@
523 \errorcontextlines\m@ne
524 \tracingnesting\z@
525 \tracingscantokens\z@
526 \tracingifs\z@
527 \tracinggroups\z@
528 \tracingparagraphs\z@
529 \tracingmacros\z@
530 \tracinglostchars\@ne
531 \tracingpages\z@
532 \tracingstats\z@
533 }%
534 \fi

535 \def\hideoutput{%
536 \tracingoutput\z@
537 \showboxbreadth\m@ne
538 \showboxdepth\m@ne
539 \tracingonline\m@ne
540 }%

541 </2ekernel | latexrelease>
542 (latexrelease)\EndIncludeInRelease
543 (latexrelease)\IncludeInRelease{0000/00/00}{\tracingnone}%
544 (latexrelease) {turn off etex tracing}%
545 (latexrelease)\let\tracingnone\@undefined
546 (latexrelease)\let\hideoutput\@undefined
547 (latexrelease)\EndIncludeInRelease

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

548 <*2ekernel>
549 \nonfrenchspacing
550 </2ekernel>

```

File c

ltx

ltvers.dtx

10 Version Identification

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

```
\fmtname
\fmtversion      1 (<*2kernel)
\patch@level     2 \def\fmtname{LaTeX2e}
                 3 \edef\fmtversion
                 4 (</2kernel)
                 5 (latexrelease)\edef\latexreleaseversion
                 6 (*2kernel| latexrelease)
                 7 {2016/03/31}
                 8 (</2kernel| latexrelease)
                 9 (*2kernel)
                10 \def\patch@level{0}
```

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

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

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

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

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

```

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

```

\IncludeInRelease

```

53 <*2ekernel | latexrelease>
54 \def\IncludeInRelease#1{\kernel@ifnextchar[%
55   {\@IncludeInRelease{#1}}
56   {\@IncludeInRelease{#1}[#1]}}
57   If a specific date has not been specified in latexrelease use '#1'.
58 \def\@IncludeInRelease#1[#2]{\@IncludeInRelease#2}
59 \def\@IncludeInRelease#1#2#3{%
60   \toks@{[#1] #3}%
61   \expandafter\ifx\csname\string#2+\@currname+IIR\endcsname\relax
62     \ifnum\expandafter\@parse@version#1//00\@nil
63       >\expandafter\@parse@version\fmtversion//00\@nil
64       \GenericInfo{}{Skipping: \the\toks@}%
65     \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
66   \else
67     \GenericInfo{}{Applying: \the\toks@}%
68   \expandafter\let\csname\string#2+\@currname+IIR\endcsname\@empty
69   \fi
70 \else
71   \GenericInfo{}{Already applied: \the\toks@}%
72   \expandafter\@gobble@IncludeInRelease
73 \fi
74 \long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{}
75 \let\EndIncludeInRelease\relax
76 </2ekernel | latexrelease>

```

File d

ltdfn.s.dtx

11 Definitions

This section contains commands used in defining other macros.

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

11.1 Initex initialisations

```
\two@digits Prefix a number less than 10 with '0'.
2 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}

\typeout Display something on the terminal.
3 \def\typeout#1{\begingroup\set@display@protect
4 \immediate\write\@unused{#1}\endgroup}

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

11.2 Saved versions of T_EX primitives

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

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

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

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

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

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

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

Therefore, macro writers should be hereby warned that these internals will probably change! It is likely that a future release of L^AT_EX will make `\-` effectively an encoding specific text command.

```

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

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

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

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

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

    17 \message{hacks,}
```

11.3 Command definitions

This section defines the following commands:

```

\@namedef    {\NAME}
Expands to \def\{\NAME}, except name can contain any characters.

\@nameuse    {\NAME}
Expands to \{\NAME}.

\@ifnextchar X{\YES}{\NO}
Expands to \YES if next character is an 'X', and to \NO otherwise. (Uses
\reserved@a-\reserved@c.) NOTE: GOBBLES ANY SPACE FOLLOWING
IT.

\@ifstar     {\YES}{\NO}
Gobbles following spaces and then tests if next the character is a '*'. If it is, then
it gobbles the '*' and expands to \YES, otherwise it expands to \NO.

\@dblarg     {\CMD}{\ARG}
Expands to \{\CMD}[\ARG]{\ARG}. Use \@dblarg\CS when \CS takes ar-
guments [ARG1]{ARG2}, where default is ARG1 = ARG2.

\@ifundefined {\NAME}{\YES}{\NO}
: If \NAME is undefined then it executes \YES, otherwise it executes \NO. More
precisely, true if \NAME either undefined or = \relax.

\@ifdefinable \NAME{\YES} Executes \YES if the user is allowed to define \NAME, otherwise
it gives an error. The user can define \NAME if \@ifundefined{NAME} is true, 'NAME'
≠ 'relax' and the first three letters of 'NAME' are not 'end', and if \endNAME is not
defined.

\newcommand  *{\FOO}[i]{TEXT}
```

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

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

`\renewcommand` $*{\langle FOO \rangle}[\langle i \rangle]{\langle TEXT \rangle}$

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

`\newenvironment` $*{\langle FOO \rangle}[\langle i \rangle]{\langle DEF1 \rangle}{\langle DEF2 \rangle}$

equivalent to:

`\newcommand{\FOO}[i]{DEF1} \def{\endFOO}{DEF2}`

(or the appropriate star forms).

`\renewenvironment`

Obvious companion to `\newenvironment`.

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

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

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

`\typeout` $\{ \langle message \rangle \}$

Produces a warning message on the terminal.

`\typein` $\{ \langle message \rangle \}$

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

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

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

`\typein`

```

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

21 \ifx\directlua\undefined

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

28 \else

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

36 \fi

```

`\@namedef`

```

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

```

`\@nameuse`

```

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

```

```

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

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

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

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

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

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

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

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

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

```

Define the actual command to be:

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

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

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

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

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

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

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

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

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

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

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

Recall that L^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`.

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

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

`\@reargdef`

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

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

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

`\renew@command`

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

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

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

```

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

```

Saved definition of \@ifdefinable.

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

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

```

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

```

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

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

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

\new@environment

```

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

```

\@newenva

```

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

```

\@newenvb

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

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

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

\renew@environment

```

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

```

\@newenv The internal version of \newenvironment.

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

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

```

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

```

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

```

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

```

`\@if`

```

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

```

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

```

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

```

`\provide@command`

```

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

```

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

```

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

```

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

```

165 \@onlypreamble\CheckCommand

```

```

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

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

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

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

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

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

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

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

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

```

11.4 Robust commands and protect

Fragile and robust commands are one of the thornier issues in L^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.
- `\@unexpandable@noexpand`, for performing a deferred write inside an `\edef`. So `\protect\foo` will write `\foo` followed by a space. If you want `\protect\foo` to be written, you should use `\@unexpandable@protect`. (Removed as never used).

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

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

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

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

which expands to:

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

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

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

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

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

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

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

```

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

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

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

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

Then we check if the macro is already robust. We do this by testing if the internal
name for a robust macro is defined, namely \foo_␣. If it is already defined do
nothing, otherwise set \foo_␣ equal to \foo and redefine \foo so that it acts like
a macro defined with \DeclareRobustCommand.
257 {%
258 \@ifundefined{\expandafter\@gobble\string#1\space}%
259 {%
260 \expandafter\let\csname
261 \expandafter\@gobble\string#1\space\endcsname=#1%
262 \edef\reserved@a{\string#1}%
263 \def\reserved@b{#1}%
264 \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
265 \edef#1{%

```

```

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

```

11.5 Internal defining commands

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

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

```

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

```

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

```

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

```

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

```

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

```

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

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

```

294 \let\kernel@ifnextchar\@ifnextchar

```


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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`\@dblarg`

```

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

```

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

```

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

```

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

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

```

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

```

319 </2ekernel>

File e

lalloc.dtx

12 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 0pt`.

`\@ne` The number 1.

`\m@ne` The number -1 .

`\tw@` The number 2.

`\sxt@@n` The number 16.

`\@m` The number 1000.

`\@MM` The number 20000.

`\@xxxii` The constant 32.
`2 \chardef\@xxxii=32`

`\@Mi` Constants 1001–1004.
`\@Mii` `3 \mathchardef\@Mi=10001`
`\@Miii` `4 \mathchardef\@Mii=10002`
`\@Miv` `5 \mathchardef\@Miii=10003`
`6 \mathchardef\@Miv=10004`

`\@tempcnta` Scratch count registers used by L^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 `\kernel`

File f

ltnctrl.dtx

13 Program control structure

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

```
1 (*2kernel)
2 \message{control,}

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

\@empty
14 \def\@empty{}

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

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

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

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

```

```

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

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

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

38 </2kernel>

```


File g

lterror.dtx

14 Error handling

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
```

14.1 General commands

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

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

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

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

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

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

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

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

```

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

```

Unfortunately T_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}%

```

`\PackageError` `\PackageWarning` `\PackageWarningNoLine` `\PackageInfo` `\ClassError` `\ClassWarning` `\ClassWarningNoLine` `\ClassInfo` These commands are intended for use by package and class writers, to give information to authors. The syntax is:

```

\PackageError{<package>}{<error>}{<help>}
\PackageWarning{<package>}{<warning>}
\PackageWarningNoLine{<package>}{<warning>}
\PackageInfo{<package>}{<info>}

```

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 be manipulated with
\setcounter (once it is defined :-)
163 \let\c@errorcontextlines\errorcontextlines
164 \c@errorcontextlines=-1

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

```

```

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

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

170 \let\@warning\@latex@warning
171 \let\@@warning\@latex@warning@no@line
172 \global\let\@latexerr\@latex@error

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

```

14.2 Specific errors

```

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

\@notdefinable Error message generated in \@ifdefinable from calls to one of the commands
\newcommand, \newlength or \newtheorem specifying an already-defined com-
mand name or one that begins \end....
187 \gdef\@notdefinable{%
188 \@latex@error{%
189 Command \@backslashchar\reserved@a\space
190 already defined.\MessageBreak
191 Or name \@backslashchar\@qend... illegal,
192 see p.192 of the manual}\@eha}

\@nolnerr Generated by \newline and \\ when called in vertical mode.
193 \gdef\@nolnerr{%
194 \@latex@error{There's no line here to end}\@eha}

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

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

```

```

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

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

\@nodocument Error produced if paragraphs are typeset in the preamble.
200 \gdef\@nodocument{%
201   \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.
202 \gdef\@badend#1{%
203   \latex@error{\protect\begin{\@currenvir}\@currencline
204     \space ended by \protect\end{#1}}\@eha}

\@badmath Called by \[, \], \( or \) when used in wrong mode.
205 \gdef\@badmath{%
206   \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.
207 \gdef\@toodeep{%
208   \latex@error{Too deeply nested}\@ehd}

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

\@badtab Called by \>, \+ , \- or \< when stepping to an undefined tab.
212 \gdef\@badtab{%
213   \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.
214 \gdef\@preamerr#1{%
215   \begingroup
216     \let\protect\relax
217     \latex@error{\ifcase #1 Illegal character\or
218       Missing @-exp\or Missing p-arg\fi\space
219       in array arg}\@ehd
220   \endgroup}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

242 %\def\@invalidchar{\@latex@error{Invalid character in input}\@ehc}
243 \</2ekernel)

```

As well as the above error commands some error messages are directly coded to save space. The Messages already present in L^AT_EX2.09 included:

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

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

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

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

File h

ltpar.dtx

15 Paragraphs

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

15.1 Implementation

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

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

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

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

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

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

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

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

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

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

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

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

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

- `@nobreak`
- `@minipage`

they should do the setting if necessary.

```
1 <*2kernel>
2 \message{par,}
```

```
\@setpar Initiate a long-term change to \par.
  \@par 3 \def\@setpar#1{\def\par{#1}\def\@par{#1}}
```

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

```
4 \def\@par{\let\par\@par\par}
5 </2kernel>
```

```
\@restorepar Restore from a short-term change to \par.
6 \def\@restorepar{\def\par{\@par}}
```

File i

ltspace.dtx

16 Spacing

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

16.1 User Commands

`\nopagebreak` [*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.

16.2 Chris' comments

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
... end of text.

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

16.3 Some immediate actions

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

- Consider replacing `\hskip 1sp` by something better (here called an ‘infinitesimal’ skip).
- Look at all `\hskip\z@` (or similar) to see if they should be changed to an ‘infinitesimal’ skip.
- Resolve the inconsistency between `\hspace` and `\hspace*`.
- Remove unnecessary `\unskips`.
- Investigate and rationalise the ‘newline’ code.
- Find better algorithms for all sorts of things or, easier(?), fix T_EX itself.

16.4 The code

```

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

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

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

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

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

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

```

```

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

```

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

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

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

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

`\@normalcr` The internal definition of the ‘normal’ definition of \ \.

```

35 \DeclareRobustCommand\ \{%
36 \let \reserved@e \relax
37 \let \reserved@f \relax
38 \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
39 \xnewline}%
40 \xnewline}
41 \expandafter\let\expandafter\@normalcr
42 \csname\expandafter\@gobble\string\ \ \endcsname

```

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

```

43 \DeclareRobustCommand\newline{\@normalcr\relax}

```

`\@xnewline`

```

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

```

`\@newline`

```

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

```

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

```

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

```


`\@getpen`

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

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

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

```
58 \def\@nbreakfalse{\global\let\if@nbreak\iffalse}
59 \def\@nbreaktrue {\global\let\if@nbreak\iftrue}
60 \@nbreakfalse
```

`\@savsk` Registers used to save the space factor and last skip.

```
\@savsf 61 \newdimen\@savsk
62 \newcount\@savsf
```

`\@bsphack` `\@esphack` 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
      \ifdim \@savsk>\z@
        \ignorespaces
      \fi
    \fi
  \fi
}
```

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

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

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

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

```

92 <latexrelease> \ifdim\@savsk>\z@
93 <latexrelease> \nobreak \hskip\z@skip
94 <latexrelease> \ignorespaces
95 <latexrelease> \fi
96 <latexrelease> \fi}%
97 <latexrelease>\EndIncludeInRelease
98 <latexrelease>\IncludeInRelease{0000/00/00}%
99 <latexrelease> {\@esphack}{hyphenation after space hack}%
100 <latexrelease>\def\@esphack{%
101 <latexrelease> \relax
102 <latexrelease> \ifhmode
103 <latexrelease> \spacefactor\@savsf
104 <latexrelease> \ifdim\@savsk>\z@
105 <latexrelease> \ignorespaces
106 <latexrelease> \fi
107 <latexrelease> \fi}%
108 <latexrelease>\EndIncludeInRelease
109 <*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
110 </2ekernel>
111 <latexrelease>\IncludeInRelease{2015/01/01}%
112 <latexrelease> {\@Esphack}{hyphenation after space hack}%
113 <*2ekernel | latexrelease>
114 \def\@Esphack{%
115 \relax
116 \ifhmode
117 \spacefactor\@savsf
118 \ifdim\@savsk>\z@
119 \nobreak \hskip\z@skip
120 \@ignoretrue
121 \ignorespaces
122 \fi
123 \fi}%
124 </2ekernel | latexrelease>
125 <latexrelease>\EndIncludeInRelease
126 <latexrelease>\IncludeInRelease{0000/00/00}%
127 <latexrelease> {\@Esphack}{hyphenation after space hack}%
128 <latexrelease>\def\@Esphack{%
129 <latexrelease> \relax
130 <latexrelease> \ifhmode
131 <latexrelease> \spacefactor\@savsf
132 <latexrelease> \ifdim\@savsk>\z@
133 <latexrelease> \@ignoretrue
134 <latexrelease> \ignorespaces
135 <latexrelease> \fi
136 <latexrelease> \fi}%
137 <latexrelease>\EndIncludeInRelease
138 <*2ekernel>

```

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

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

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

16.5 Vertical spacing

L^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.

```

139 \def\@xaddvskip{%
140   \ifdim\lastskip<\@tempskipb
141     \vskip-\lastskip
142     \vskip\@tempskipb
143   \else
144     \ifdim\@tempskipb<\z@
145       \ifdim\lastskip<\z@
146         \else
147           \advance\@tempskipb\lastskip
148           \vskip-\lastskip
149           \vskip \@tempskipb
150         \fi
151       \fi
152     \fi}

```

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

```

153 \def\addvspace#1{%
154   \ifvmode
155     \if@minipage\else
156       \ifdim \lastskip =\z@
157         \vskip #1\relax
158       \else
159         \@tempskipb#1\relax
160         \@xaddvskip
161       \fi
162     \fi
163   \else
164     \@noitemerr
165   \fi}

```

`\addpenalty`

```

166 </2ekernel>
167 <latexrelease>\IncludeInRelease{2015/01/01}%
168 <latexrelease>          {\addpenalty}{\addpenalty}%
169 <*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.

```

170 \def\addpenalty#1{%
171   \ifvmode
172     \if@minipage
173     \else
174       \if@nobreak
175       \else
176         \ifdim\lastskip=\z@
177           \penalty#1\relax
178         \else
179           \@tempskipb\lastskip

```

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

```

180         \begingroup
181         \@tempskipa\@tempskipb
182         \advance \@tempskipb
183         \ifdim\prevdepth>\maxdepth\maxdepth\else
If \prevdepth is -1000pt due to \nointerlineskip we better not add it!
184         \ifdim \prevdepth = -\@m\p@ \z@ \else \prevdepth \fi
185         \fi
186         \vskip -\@tempskipb
187         \penalty#1%
188         \ifdim\@tempskipa=\@tempskipb
Do nothing if the \prevdepth check made no adjustment.
189         \else
Combine the prevdepth adjustment into a single skip.
190         \advance\@tempskipb -\@tempskipa
191         \vskip \@tempskipb
192         \fi
The final skip is always the specified length.
193         \vskip \@tempskipa
194     \endgroup
195     \fi
196     \fi
197     \fi
198 \else
199     \@noitemerr
200 \fi}%
201 </2ekernel | latexrelease>
202 <latexrelease>\EndIncludeInRelease
203 <latexrelease>\IncludeInRelease{0000/00/00}%
204 <latexrelease>     {\addpenalty}{\addpenalty}%
205 <latexrelease>\def\addpenalty#1{%
206 <latexrelease> \ifvmode
207 <latexrelease>     \if@minipage
208 <latexrelease>     \else
209 <latexrelease>         \if@nobreak
210 <latexrelease>         \else
211 <latexrelease>             \ifdim\lastskip=\z@
212 <latexrelease>                 \penalty#1\relax
213 <latexrelease>             \else
214 <latexrelease>                 \@tempskipb\lastskip
215 <latexrelease>                 \vskip -\lastskip
216 <latexrelease>                 \penalty#1%
217 <latexrelease>                 \vskip\@tempskipb
218 <latexrelease>             \fi
219 <latexrelease>         \fi
220 <latexrelease>     \fi
221 <latexrelease> \else
222 <latexrelease>     \@noitemerr
223 <latexrelease> \fi}%
224 <latexrelease>\EndIncludeInRelease
225 <*/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.

```

226 \DeclareRobustCommand\vspace{\@ifstar\@vspacer\vspace}
227 \def\@vspace #1{%
228   \ifvmode
229     \vskip #1
230     \vskip\z@skip
231   \else
232     \@bsphack
233     \vadjust{\@restorepar
234               \vskip #1
235               \vskip\z@skip
236             }%
237     \@esphack
238   \fi}

239 \def\@vspacer#1{%
240   \ifvmode
241     \dimen@\prevdepth
242     \hrule \@height\z@
243     \nobreak
244     \vskip #1
245     \vskip\z@skip
246     \prevdepth\dimen@
247   \else
248     \@bsphack
249     \vadjust{\@restorepar
250               \hrule \@height\z@
251               \nobreak
252               \vskip #1
253               \vskip\z@skip}%
254     \@esphack
255   \fi}

```

```

\smallskip
\medskip 256 \def\smallskip{\vspace\smallskipamount}
\bigskip 257 \def\medskip{\vspace\medskipamount}
258 \def\bigskip{\vspace\bigskipamount}

```

```

\smallskipamount
\medskipamount 259 \newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
\bigskipamount 260 \newskip\medskipamount \medskipamount =6pt plus 2pt minus 2pt
261 \newskip\bigskipamount \bigskipamount =12pt plus 4pt minus 4pt

```

16.6 Horizontal space (and breaks)

`\nbreakdashes` 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!

```

262 \DeclareRobustCommand{\nobreakdashes}{%
263   \leavevmode
264   \toks@{}}%
265 \def\reserved@a##1{\toks@\expandafter{\the\toks@-}%
266   \futurelet\@let@token \reserved@b}%
267 \def\reserved@b  {\ifx\@let@token -%
268   \expandafter\reserved@a
269   \else
270   \setbox\z@ \hbox{\the\toks@\nobreak}%
271   \unhbox\z@
272   \spacefactor\sfcode'\-
273   \fi}%
274 \futurelet\@let@token \reserved@b
275 }

```

`\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` .

```

276 \DeclareRobustCommand{\nobreakspace}{%
277   \leavevmode\nobreak\ }
278 \catcode '\~ =13
279 \def~{\nobreakspace{}}
280 \expandafter\let\expandafter\@xobeysp\csname nobreakspace \endcsname

```

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

```

281 \DeclareRobustCommand{\,}{%
282   \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi
283 }

```

`\@` 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.

```

284 </2ekernel>
285 <latexrelease>\IncludeInRelease{2015/01/01}%
286 <latexrelease>   {\@}{Space after \@}%
287 <*2ekernel | latexrelease>

```



```

288 \def\@{\spacefactor\@m{}}%
289 </2ekernel | latexrelease)
290 (latexrelease)\EndIncludeInRelease
291 (latexrelease)\IncludeInRelease{0000/00/00}%
292 (latexrelease)          {\@}{Space after \@}%
293 (latexrelease)\def\@{\spacefactor\@m}%
294 (latexrelease)\EndIncludeInRelease
295 (*2ekernel)

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

\@hspace
297 \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
298 \def\@hspacer#1{\vrule \@width\z@\nobreak
299           \hskip #1\hskip \z@skip}

\fill
300 \newskip\fill
301 \fill = Opt plus 1fill

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

\thinspace
\negthinspace 303 \def\thinspace{\kern .16667em }
\enspace      304 \def\negthinspace{\kern-.16667em }
              305 \def\enspace{\kern.5em }

\enskip
\quad        306 \def\enskip{\hskip.5em\relax}
\qquad      307 \def\quad{\hskip1em\relax}
\qqquad     308 \def\qqquad{\hskip2em\relax}

\obeycr      The following definitions will probably get deleted or moved to compatibility mode
\restorecr   soon.
309 {\catcode'\^^M=13 \gdef\obeycr{\catcode'\^^M13 \def^^M{\\}\relax}%
310   \gobblecr}%
311 {\catcode'\^^M=13 \gdef\gobblecr{\@ifnextchar
312 \gobble\ignorespaces}}
313 \gdef\restorecr{\catcode'\^^M5 }}

314 </2ekernel)

```

File j

ltlogos.dtx

17 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 <*/2ekernel)
2 \def\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 $\LaTeX 2_{\epsilon}$ logo as proposed by A-W designers.

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

File k

lfiles.dtx

18 File Handling

The following user commands are defined in this part:

<code>\document</code>	<code>(ie \begin{document})</code>	Reads in the .AUX files and <code>\catcode</code> 's @ to 12.
<code>\nofiles</code>		Suppresses all file output by setting <code>\@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> .

```
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

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

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

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

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

\document Cancel the \@begingroup from \@begin
           11 \def\document{\endgroup

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

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

```

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

```

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

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

So that punctuation in headings is not disturbed by verbatim or other local changes to the space factor codes, save the document default here. This will be locally reset by the output routine. For special cases a class may want to define `\normalsfcodes` directly, in case that definition will be used. (This is an old bug, problem existed in $\text{\LaTeX}2.0x$ and plain \TeX .)

```

38 \ifx\normalsfcodes@empty
39   \ifnum\sfcode'\.=\@m
40     \let\normalsfcodes\frenchspacing
41   \else
42     \let\normalsfcodes\nofrenchspacing
43   \fi
44 \fi
```

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

```

45 \noskipsecfalse
46 \let \@refundefined \relax
```

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

```

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

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

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

```

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

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

```

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

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

```
58 \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.

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

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

```
60 \ignorespaces}
```

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

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

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

`\nofiles` Set `\@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.

```
63 \def\nofiles{%
```

```
64 \@fileswfalse
```

```
65 \typeout{No auxiliary output files.^^J}%
```

```
66 \long\def\protected@write##1##2##3%
```

```
67 {\write\m@ne{}\if@nobreak\ifvmode\nobreak\fi\fi}%
```

```
68 \let\makeindex\relax
```

```
69 \let\makeglossary\relax}
```

```
70 \@onlypreamble\nofiles
```

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

```
71 \long\def \protected@write#1#2#3{%
```

```
72 \begingroup
```

```
73 \let\thepage\relax
```

```
74 #2%
```

```
75 \let\protect\@unexpandable@protect
```

```
76 \edef\reserved@a{\write#1{#3}}%
```

```
77 \reserved@a
```

```
78 \endgroup
```

```
79 \if@nobreak\ifvmode\nobreak\fi\fi
```

```
80 }
```

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

`\includeonly`

```
82 \def\includeonly#1{%
```

```
83 \@partswtrue
```

```

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

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

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

\@include

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

\@writeckpt

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

```



```

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

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

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

\@charlb The following defines \@charlb and \@charrb to be { and }, respectively with
\@charrb \catcode 11.
129 {\catcode' [=1 \catcode' ]=2
130 \catcode' {=11 \catcode' }=11
131 \gdef\@charlb[{
132 \gdef\@charrb[}]
133 ]% }brace matching

```

18.1 Safe Input Macros

```

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

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

```

```

\InputIfFileExists Now define \InputIfFileExists to input #1 if it seems to exist. Immediately
prior to the input, #2 is executed. If the file #1 does not exist, execute '#3'.
160 \long\def \InputIfFileExists#1#2{%
161   \IfFileExists{#1}%
162     {#2\@addtofilelist{#1}\@input \@filef@und}}

\input Input a file: if the argument is given in braces use safe input macros, otherwise
use TEX's primitive \input command (which is called \@input in LATEX).
163 \def\input{\@ifnextchar\bgroup\iinput\@input}

\iinput Define \@iinput (i.e., \input) in terms of \InputIffileExists.
164 \def\@iinput#1{%
165   \InputIfFileExists{#1}{}%
166   {\filename@parse{#1}%
167     \edef\reserved@a{\noexpand\@missingfileerror
168       {\filename@area\filename@base}%
169       {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
170   \reserved@a}}

\@input Define \@input in terms of \IffileExists. So this is a 'safe input' command,
but the files input are not listed by \listfiles.
We don't want .aux, .toc files etc be listed by \listfiles. However, some-
thing like .bbl probably should be listed and thus should be implemented not by
\@input.
171 \def\@input#1{%
172   \IfFileExists{#1}{\@input\@filef@und}{\typeout{No file #1.}}}

\@input@ Version of \@input that does add the file to \@filelist.
173 \def\@input@#1{\InputIfFileExists{#1}{\typeout{No file #1.}}}

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

```

`\@obsoletefile` For compatibility with L^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.

```

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

```

18.2 Listing files

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

```

199 \let\@filelist\@gobble

```

`\@addtofilelist` Add to the list of files input so far. This ‘real’ definition is only used for ‘cfg’ files during `initex`. An initial definition of `\@gobble` has already been set.

```

200 %\def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}

```

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

```

201 \def\listfiles{%
202   \let\listfiles\relax
203   \def\@listfiles##1##2##3##4##5##6##7##8##9\@{%
204     \def\reserved@d{\}%
205     \@tfor\reserved@c:=##1##2##3##4##5##6##7##8\do{%
206       \ifx\reserved@c\reserved@d
207         \edef\filename@area{ \filename@area}%
208       \fi}}%
209   \def\@dofilelist{%
210     \typeout{^^J *File List*}%
211     \@for\@currname:=\@filelist\do{%
212       \filename@parse\@currname
213       \edef\reserved@a{%
214         \filename@base.%
215         \ifx\filename@ext\relax tex\else\filename@ext\fi}%
216       \expandafter\let\expandafter\reserved@b
217         \csname ver@\reserved@a\endcsname
218       \expandafter\expandafter\expandafter\@listfiles\expandafter
219         \filename@area\filename@base\@
220       \typeout{%
221         \filename@area\reserved@a
222         \ifx\reserved@b\relax\else\@spaces\reserved@b\fi}}%
223     \typeout{ *****^^J}}

```

The `\@filelist` will be de-activated if `\listfiles` does not appear in the preamble. `\begin{document}` contains code equivalent to the following:

```

\AtBeginDocument{%
  \ifx\@listfiles\undefined
    \let\@filelist\relax
    \let\@addtofilelist\@gobble
  \fi}
224 \@onlypreamble\listfiles

```

\@dofilelist

225 \let\@dofilelist\relax

226 \endkernel

File 1

ltoutenc.dtx

19 Font encodings

This section of the kernel contains commands for declaring encoding-specific commands, such as accents. It also contains the code for some of the encoding files, including `omlenc.def`, `omsenc.def`, `t1enc.def` and `ot1enc.def` files, which define the OLM, OMS, T1 and OT1 encodings, and the `fontenc` package for selecting encodings.

The `fontenc` package has options for encodings, of which the last option is the default encoding. For example, to use the OT2, OT3 and T1 encodings, with T1 as the default, you say:

```
\usepackage[OT2,OT3,T1]{fontenc}
```

The standard kernel set-up loads font encoding files and selects an encoding as follows.

```
\input {omlenc.def}
\input {t1enc.def}
\input {ot1enc.def}
\input {omsenc.def}
\fontencoding{OT1}
```

Note that the files in the standard `inputenc` package depend on this behaviour of the kernel.

The syntax for declaring encoding-specific commands is:

```
\DeclareTextCommand{<command>}{<encoding>}
                        [ <number> ] [ <default> ] { <commands> }
```

This command is like `\newcommand`, except that it defines a command which is specific to one encoding. The resulting command is always robust, even if its definition is fragile. For example, the definition of `\l` in the OT1 encoding is:

```
\DeclareTextCommand{\l}{OT1}{\@xxxii l}
```

`\DeclareTextCommand` takes the same optional arguments as `\newcommand`.

```
\ProvideTextCommand{<command>}{<encoding>}
                        [ <number> ] [ <default> ] { <commands> }
```

This acts like `\DeclareTextCommand`, but does nothing if the command is already defined.

```
\DeclareTextSymbol{<command>}{<encoding>}{<slot>}
```

This command defines a text symbol, with a particular slot in that encoding. The commands:

```
\DeclareTextSymbol{\ss}{OT1}{25}
\DeclareTextCommand{\ss}{OT1}{\char25 }
```

have the same effect, but the `\DeclareTextSymbol` is faster.

```
\DeclareTextAccent{<command>}{<encoding>}{<slot>}
```

This command declares a text accent. The commands:

```
\DeclareTextAccent{"}{OT1}{127}
\DeclareTextCommand{"}{OT1}{\add@accent {127}}
```

have the same effect.

```
\DeclareTextComposite{<command>
                    {<encoding>}{<argument>}{<slot>}
```

This command declares a composite letter, for example in the T1 encoding `\'a` is slot 225, which is declared by:

```
\DeclareTextComposite{\'}{T1}{a}{225}
```

The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

`\DeclareTextComposite` is the most common example of using the more general declaration `\DeclareTextCompositeCommand`, which can define a composite to be an arbitrary piece of text.

```
\DeclareTextCompositeCommand{<command>
                             {<encoding>}{<argument>}{<text>}
```

For example, in the OT1 encoding `Å` has a hand-crafted definition this is declared as follows

```
\DeclareTextCompositeCommand{\r}{OT1}{A}
{\leavevmode\setbox\z@\hbox{!}\dimen@ \ht\z@\advance\dimen@-1ex%
 \rlap{\raise.67\dimen@\hbox{\char23}}A}
```

The *command* will normally have been declared with `\DeclareTextAccent`, or as a one-argument `\DeclareTextCommand`.

The commands defined using the above declarations can be used in two ways. Normally they are used by just calling the command in the appropriate encoding, for example `\ss`. However, sometimes you may wish to use a command in an encoding where it is not defined. If the command has no arguments, then you can use it in another encoding by calling `\UseTextSymbol`:

```
\UseTextSymbol{<encoding>}{<command>}
```

For example, `\UseTextSymbol{OT1}{\ss}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\ss}
```

If the command has one argument then you can use it in another encoding by calling `\UseTextAccent`:

```
\UseTextAccent{<encoding>}{<command>}{<text>}
```

For example, if the current encoding is OT2 then `\UseTextAccent{OT1}{\'}{a}` has the same effect as:

```
{\fontencoding{OT1}\selectfont\'}{\fontencoding{OT2}\selectfont a}
```

You can also declare a default definition for a text command, which will be used if the current encoding has no appropriate definition. Such use will also set the definition for this command in the current encoding to equal this default definition; this makes subsequent uses of the command much faster.

```
\DeclareTextCommandDefault{<command>}{<definition>}
```

For example, the default definition of the command `\textonequarter` (which produces the fraction $\frac{1}{4}$) could be built using math mode:

```
\DeclareTextCommandDefault{\textonequarter}{\ensuremath {\frac{1}{4}}}
```

There is a matching `\Provide` command which will not override an existing default definition:

```
\ProvideTextCommandDefault{<command>}{<definition>}
```

The most common use for these commands is to use symbols from other encodings, so there are some optimizations provided:

```
\DeclareTextSymbolDefault{<command>}{<encoding>}
\DeclareTextAccentDefault{<command>}{<encoding>}
```

are short for:

```
\DeclareTextCommandDefault{<command>}
      {\UseTextSymbol{<encoding>}{<command>}}
\DeclareTextCommandDefault[1]{<command>}
      {\UseTextAccent{<encoding>}{<command>}{#1}}
```

For example, to make OT1 the default encoding for `\ss` and `\'` you say:

```
\DeclareTextSymbolDefault{\ss}{OT1}
\DeclareTextAccentDefault{\'}{OT1}
```

Note that you can use these commands on any zero- or one-argument commands declared with `\DeclareText*` or `\ProvideText*`, not just those defined using `\DeclareTextSymbol` or `\DeclareTextAccent`.

19.1 Removing encoding-specific commands

In some cases encoding definitions are given to provide some limited support since nothing better is available, for example, the definition for `\textdollar` in OT1 is a hack since \$ and £ actually share the same slot in this encoding. Thus if such a glyph becomes available in a different encoding (e.g., TS1) one would like to get rid of the 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}
```

19.2 The order of declarations

If an encoding-specific command is defined for more than one encoding, then it will execute fastest in the encoding in which it was defined last since its top-level definition will be set up to execute in that encoding without any overhead.

For this reason the file `fonttext.ltx` currently first loads the definitions for the T1 encoding and then those for the OT1 encoding so that typesetting in OT1 is optimized since that is (still) the default. However, when T1 is explicitly requested (via `\usepackage[T1]{fontenc}`) the top-level definitions are automatically changed to favour T1 since its declarations are reloaded in the process.

For the same reason default declarations should never come last since they are implemented as a special encoding themselves (with the name ?). Specifying them last would simply mean to make those encoding-specific commands equally inefficient in all encodings. Therefore the `textcomp` package, for example, first sets up all defaults to point to TS1 and then declares the commands in the TS1 encoding.

19.3 Docstrip modules

This `.dtx` file is be used to generate several related files containing font encoding definitions. The mutually exclusive docstrip options are listed here.

T1	generates <code>t1enc.def</code> for the Cork encoding.
TS1	generates <code>ts1enc.def</code> for the Text Companion encoding.
TS1sty	generates <code>textcomp.sty</code> , package that sets up use of the Text Companion encoding.
OT1	generates <code>ot1enc.def</code> for Knuth's CM encoding.
OMS	generates <code>omsenc.def</code> for Knuth's math symbol encoding.
OML	generates <code>omlenc.def</code> for Knuth's math letters encoding.
OT4	generates <code>ot4enc.def</code> for the Polish extension to the OT1 encoding, created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete.
package	generates <code>fontenc.sty</code> for selecting encodings.
2kernel	for the kernel commands.

19.4 Definitions for the kernel

19.4.1 Declaration commands

This section contains definitions for commands such as accents which depend on the current encoding. These commands will usually be kept in `.def` files, for

example `ot1enc.def` contains the definitions for the OT1 encoding.

```
1 (*2kernel)
2 \message{font encodings,}
   Far too many macros in one block here!
```

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

If you say:

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

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

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

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

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

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

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

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

- If `\protect` is `\@typeset@protect` and `\cf@encoding` is T1, then we execute `\T1\foo`. This should be the normal behaviour, and is optimized for speed.
- If `\protect` is `\@typeset@protect`, `\cf@encoding` is (say) OT1, and `\OT1\foo` is defined, then we execute `\OT1\foo`.

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

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

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

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

```

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

31 \def\@changed@cmd#1#2{%
32   \ifx\protect\@typeset@protect
33     \@inmathwarn#1%
34     \expandafter\ifx\csname\cf@encoding\string#1\endcsname\relax
35       \expandafter\ifx\csname ?\string#1\endcsname\relax
36         \expandafter\def\csname ?\string#1\endcsname{%
37           \TextSymbolUnavailable#1%
38         }%
39       \fi
40       \global\expandafter\let
41         \csname\cf@encoding\string#1\expandafter\endcsname
42         \csname ?\string#1\endcsname
43       \fi
44       \csname\cf@encoding\string#1%
45         \expandafter\endcsname
46   \else
47     \noexpand#1%
48   \fi}

49 \gdef\TextSymbolUnavailable#1{%
50   \@latex@error{%
51     Command \protect#1 unavailable in encoding \cf@encoding%
52   }\@eha}

```

The command `\@inmathwarn` produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't

call `\relax` before the `\ifmmode`. This means that it is possible for the warning to fail to be issued at the beginning of a row of an `halign` whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a `\relax` would break ligatures and kerning between text symbols.

A more efficient solution would be to make `\@inmathwarn` and `\@inmatherr` equal to `\@empty` and `\relax` by default, and to have `\everymath` reset them to their usual definitions. This is left for future investigation (for example it may break some third party code).

```
53 \def\@inmathwarn#1{%
54   \ifmmode
55     \@latex@warning{Command \protect#1 invalid in math mode}%
56   \fi}
```

`\DeclareTextCommandDefault` These define commands with encoding ?.

`\ProvideTextCommandDefault` Note that `\DeclareTextCommandDefault` can only be used in the preamble, but that the `\Provide` version is allowed in inputenc `.def` files, so is allowed anywhere.

```
57 \def\DeclareTextCommandDefault#1{%
58   \DeclareTextCommand#1?}

59 \def\ProvideTextCommandDefault#1{%
60   \ProvideTextCommand#1?}

61 \@onlypreamble\DeclareTextCommandDefault
62 %\@onlypreamble\ProvideTextCommandDefault
```

They require `\?-cmd` to be initialized as `\@changed@cmd`.

```
63 \expandafter\let\csname?-cmd\endcsname\@changed@cmd
```

`\DeclareTextAccent` This is just a disguise for defining a T_EX `\accent` command.

```
64 \def\DeclareTextAccent#1#2#3{%
65   \DeclareTextCommand#1{#2}{\add@accent{#3}}
66 \@onlypreamble\DeclareTextAccent
```

`\add@accent` To save space this code is shared between all text accents that are set using the `\accent` primitive. The argument is pre-set in a box so that any font loading that is needed is already done within the box. This is needed because font-loading involves grouping and that would prevent the accent mechanism from working so that the accent would not be positioned over the argument. Declarations that change the font should be allowed (only low-level ones are at present) inside the argument of an accent command, but not size changes, as they involve `\setbox` operations which also inhibit the mechanism of the `\accent` primitive.

Note that the whole process is within a group. For a detailed discussion of this reimplementation and its deficiencies, see pr/3160.

```
67 \def\add@accent#1#2{\hmode@bgroup
```

Turn off the group in `\UseTextSymbol` in case this is used inside the argument of `\add@accent`.

```
68   \let\hmode@start@before@group\@firstofone
69   \setbox\@tempboxa\hbox{#2%
```

When presetting the argument in a box we record its `\spacefactor` for later use after the accent got typeset. This way something like `\'A` gets the spacefactor of A (i.e., 999) rather than the default value of 1000.

```
70     \global\mathchardef\accent@spacefactor\spacefactor}%
71     \accent#1 #2\egroup\spacefactor\accent@spacefactor}
```

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

```
72 \let\accent@spacefactor\relax
```

```
\hmode@bgroup
```

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

`\DeclareTextCompositeCommand` Another amusing game to play with `\expandafter`, `\csname`, and `\string`. When you say `\DeclareTextCompositeCommand{\foo}{T1}{a}{bar}`, we look to see if the expansion of `\T1\foo` begins with `\@text@composite`, and if it doesn't, we redefine `\T1\foo` to be:

```
\@text@composite@x
\@strip@args
#1 -> \@text@composite \T1\foo #1\@empty \@text@composite {...}
```

where `...` is the previous definition of `\T1\foo`. Finally, we define `\T1\foo-a` to expand to `bar`.

```
74 \def\DeclareTextCompositeCommand#1#2#3#4{%
75   \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
76   \expandafter\expandafter\expandafter\ifx
77   \expandafter\@car\reserved@a\relax\relax\@nil \@text@composite \else
78     \edef\reserved@b##1{%
79       \def\expandafter\noexpand
80         \csname#2\string#1\endcsname###1{%
81         \noexpand\@text@composite
82         \expandafter\noexpand\csname#2\string#1\endcsname
83         ###1\noexpand\@empty\noexpand\@text@composite
84         {##1}}}%
85   \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
86   \fi
87   \expandafter\def\csname\expandafter\string\csname
88     #2\endcsname\string#1-\string#3\endcsname{#4}}
89 \@onlypreamble\DeclareTextCompositeCommand
```

This all works because:

```
\@text@composite \T1\foo A\@empty \@text@composite {...}
```

expands to `\T1\foo-A` if `\T1\foo-A` has been defined, and `{...}` otherwise.

Note that `\@text@composite` grabs the first token of the argument and puts just that in the 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.

```
90 \def\@text@composite#1#2#3\@text@composite{%
91   \expandafter\@text@composite@x
92     \csname\string#1-\string#2\endcsname}
```

Originally the `\@text@composite@x` macro had two arguments and if #1 was not `\relax` it was executed, otherwise #2 was executed. All this happened within the `\ifx` code so that neither #1 nor #2 could have picked up any additional arguments from the input stream. This has now been changed using the typical `\@firstoftwo / \@secondoftwo` coding. This way the final expansion will happen without any `\else` or `\fi` intervening in the case that we need to get a further token from the input stream.

```
93 \def\@text@composite@x#1{%
94   \ifx#1\relax
95     \expandafter\@secondoftwo
96   \else
97     \expandafter\@firstoftwo
98   \fi
99   #1}
```

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

```
100 \catcode\z@=11\relax
101 \def\DeclareTextComposite#1#2#3#4{%
102   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
103   \bgroup
104     \lccode\z@#4%
105     \lowercase{%
106   \egroup
107     \reserved@a ^^@}}
108 \catcode\z@=15\relax
109 \@onlypreamble\DeclareTextComposite
```

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

For a detailed discussion of this reimplementaion and its deficiencies, see pr/3160.

```

110 \def\UseTextAccent#1#2#3{%
111   \hmode@start@before@group
112   {%

```

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

```

113   \let\hmode@start@before@group\@firstofone
114   \let\@curr@enc\cf@encoding
115   \@use@text@encoding{#1}%
116   #2{\@use@text@encoding\@curr@enc#3}%
117   }}

```

```

118 \def\UseTextSymbol#1#2{%
119   \hmode@start@before@group
120   {%
121     \def\@wrong@font@char{\MessageBreak
122       for \noexpand\symbol'\string#2'}%
123     \@use@text@encoding{#1}%
124     #2%
125   }%
126   }

```

```

127 \def\@use@text@encoding#1{%
128   \edef\font@encoding{#1}%
129   \xdef\font@name{%
130     \csname\curr@fontshape/\font@size\endcsname}%
131   \pickup@font
132   \font@name
133   \@@enc@update}

```

`\hmode@start@before@group` The `\hmode@start@before@group` starts hmode and should be immediately followed by an explicit `{...}`. Its purpose is to ensure that hmode is started before this group is opened. Inside `\add@accent` and `\UseTextAccent` it is redefined to remove this group so that it doesn't conflict with the `\accent` primitive.

For a detailed discussion see pr/3160.

```

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

```

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

```

\DeclareTextAccentDefault
135 \def\DeclareTextSymbolDefault#1#2{%
136   \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}
137 \def\DeclareTextAccentDefault#1#2{%
138   \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}
139 \@onlypreamble\DeclareTextSymbolDefault
140 \@onlypreamble\DeclareTextAccentDefault

```

`\UndeclareTextCommand` This command safely removes an encoding specific declaration for a given encoding. It is helpful if one intends to use the default definition always and therefore wants to get rid of a declaration for some specific encoding.

```

141 \def\UndeclareTextCommand#1#2{%

```

If there is no declaration for the current encoding do nothing. (This makes a hash table entry but without eTeX we can't do anything about that).

```

142   \expandafter\ifx\csname#2\string#1\endcsname\relax
143   \else

```

Else: throw away that declaration.

```
144 \global\expandafter\let\csname#2\string#1\endcsname
145 \undefined
```

But this is unfortunately not enough, we have to take a look at the top-level definition of the encoding specific command which for a command `\foo` would look similar to `\T1-cmd \foo \T1\foo` (three tokens).

Of course, instead of `T1` one could see a different encoding name; which one depends the encoding for which `\foo` was declared last.

Now assume we have just removed the declaration for `\foo` in `T1` and the top-level of `\foo` expands to the above. Then we better change that pretty fast otherwise we do get an “undefined csname error” when we try to typeset `\foo` within `T1` instead of getting the default definition for `\foo`. And what is the best way to change that top-level definition? Well, the only “encoding” we know for sure will still be around is the default encoding denoted by `?`.

Thus in case the last token of the top-level expansion is now undefined we change the declaration to look like `\?-cmd \foo \?\foo` which is done by the following (readable?) code:

```
146 \expandafter\expandafter\expandafter
147 \ifx\expandafter\@thirdofthree#1\undefined
148 \expandafter\gdef\expandafter#1\expandafter
149 {\csname ?-cmd\expandafter\endcsname\expandafter
150 #1\csname?\string#1\endcsname}%
151 \fi
152 \fi
153 }
154 \@onlypreamble\UndeclareTextCommand
```

19.4.2 Hyphenation

```
\patterns We redefine \patterns and \hyphenation to allow the use of commands declared
\@patterns with \DeclareText* to be used inside them.
\hyphenation 155 %\let\@patterns\patterns
\@hyphenation 156 %\let\@hyphenation\hyphenation
157 %\def\patterns{%
158 % \bgroup
159 % \let\protect\@empty
160 % \let\@typeset@protect\@empty
161 % \let\@changed@x\@changed@x@mouth
162 % \afterassignment\egroup
163 % \@patterns
164 %}
165 %\def\hyphenation{%
166 % \bgroup
167 % \let\protect\@empty
168 % \let\@typeset@protect\@empty
169 % \let\@changed@x\@changed@x@mouth
170 % \afterassignment\egroup
171 % \@hyphenation
172 %}
```

19.4.3 Miscellanea

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

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

```
173 \def\@tabacckludge#1{\expandafter\@changed@cmd
174                               \csname\string#1\endcsname\relax}
175 \let\a=\@tabacckludge
```

19.4.4 Default encodings

We define the default encodings for most commands to be either OT1, OML or OMS. These defaults are in the kernel and therefore fonts with these encodings must be available unless these defaults are redefined elsewhere. Recall that the standard kernel loads the encoding files for these encodings, and also that for the T1 encoding.

The naming conventions in the kernel are not what we would use if we were starting from scratch... Those defined by DEK (like `\ae` and `\ss`) or by the T_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:

```
176 \DeclareTextAccentDefault{"}{OT1}
177 \DeclareTextAccentDefault{'}{OT1}
178 \DeclareTextAccentDefault{.}{OT1}
179 \DeclareTextAccentDefault{=}{OT1}
180 \DeclareTextAccentDefault{H}{OT1}
181 \DeclareTextAccentDefault{^}{OT1}
182 \DeclareTextAccentDefault{'}{OT1}
183 \DeclareTextAccentDefault{b}{OT1}
184 \DeclareTextAccentDefault{c}{OT1}
185 \DeclareTextAccentDefault{d}{OT1}
186 \DeclareTextAccentDefault{r}{OT1}
187 \DeclareTextAccentDefault{u}{OT1}
188 \DeclareTextAccentDefault{v}{OT1}
189 \DeclareTextAccentDefault{~}{OT1}
```

Some symbols from OT1:

```
190 %\DeclareTextSymbolDefault{AA}{OT1}
191 \DeclareTextSymbolDefault{AE}{OT1}
192 \DeclareTextSymbolDefault{L}{OT1}
193 \DeclareTextSymbolDefault{OE}{OT1}
194 \DeclareTextSymbolDefault{O}{OT1}
195 %\DeclareTextSymbolDefault{aa}{OT1}
```



```

196 \DeclareTextSymbolDefault{\ae}{OT1}
197 \DeclareTextSymbolDefault{\i}{OT1}
198 \DeclareTextSymbolDefault{\j}{OT1}

199 \DeclareTextSymbolDefault{\ij}{OT1}
200 \DeclareTextSymbolDefault{\IJ}{OT1}

201 \DeclareTextSymbolDefault{\l}{OT1}
202 \DeclareTextSymbolDefault{\oe}{OT1}
203 \DeclareTextSymbolDefault{\o}{OT1}
204 \DeclareTextSymbolDefault{\ss}{OT1}
205 \DeclareTextSymbolDefault{\textdollar}{OT1}
206 \DeclareTextSymbolDefault{\textemdash}{OT1}
207 \DeclareTextSymbolDefault{\textendash}{OT1}
208 \DeclareTextSymbolDefault{\textexclamdown}{OT1}
209 %\DeclareTextSymbolDefault{\textthyphenchar}{OT1}
210 %\DeclareTextSymbolDefault{\textthyphen}{OT1}
211 \DeclareTextSymbolDefault{\textquestiondown}{OT1}
212 \DeclareTextSymbolDefault{\textquotedblleft}{OT1}
213 \DeclareTextSymbolDefault{\textquotedblright}{OT1}
214 \DeclareTextSymbolDefault{\textquoteleft}{OT1}
215 \DeclareTextSymbolDefault{\textquoteright}{OT1}
216 \DeclareTextSymbolDefault{\textsterling}{OT1}

```

Some symbols from OMS:

```

217 \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
218 \DeclareTextSymbolDefault{\textbackslash}{OMS}
219 \DeclareTextSymbolDefault{\textbar}{OMS}
220 \DeclareTextSymbolDefault{\textbardbl}{OMS}
221 \DeclareTextSymbolDefault{\textbraceleft}{OMS}
222 \DeclareTextSymbolDefault{\textbraceright}{OMS}
223 \DeclareTextSymbolDefault{\textbullet}{OMS}
224 \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
225 \DeclareTextSymbolDefault{\textdagger}{OMS}
226 \DeclareTextSymbolDefault{\textparagraph}{OMS}
227 \DeclareTextSymbolDefault{\textperiodcentered}{OMS}
228 \DeclareTextSymbolDefault{\textsection}{OMS}
229 \DeclareTextAccentDefault{\textcircled}{OMS}

```

Some symbols from OML:

```

230 \DeclareTextSymbolDefault{\textless}{OML}
231 \DeclareTextSymbolDefault{\textgreater}{OML}
232 \DeclareTextAccentDefault{\t}{OML}

```

Some defaults we can fake.

The interface for defining `\copyright` changed, it used to use `\expandafter` to add braces at the appropriate points.

```

233 \DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
234 % \expandafter\def\expandafter
235 % \copyright\expandafter{\expandafter{\copyright}}

236 \DeclareTextCommandDefault{\textasciicircum}{\~{}}
237 \DeclareTextCommandDefault{\textasciitilde}{\~{}}
238 \DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
239 \DeclareTextCommandDefault{\textunderscore}{%
240 \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

```

```

241 \DeclareTextCommandDefault{\textvisiblespace}{%
242   \mbox{\kern.06em\vrule \@height.3ex}%
243   \vbox{\hrule \@width.3em}%
244   \hbox{\vrule \@height.3ex}}

```

Using `\fontdimen3` in the next definition is some sort of a kludge (since it is the interword stretch) but it makes the ellipsis come out right in mono-spaced fonts too (since there it is zero).

```

245 \DeclareTextCommandDefault{\textellipsis}{%
246   .\kern\fontdimen3\font
247   .\kern\fontdimen3\font
248   .\kern\fontdimen3\font}

249 %\DeclareTextCommandDefault{\textregistered}{\textcircled{\scshape r}}
250 \DeclareTextCommandDefault{\textregistered}{\textcircled{%
251   \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
252 \DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
253 \DeclareTextCommandDefault{\SS}{SS}

254 \DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
255 \DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}

```

19.4.5 Math material

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

```

256 \DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
257 \DeclareRobustCommand{\{ }{\ifmmode\lbrace\else\textbraceleft\fi}
258 \DeclareRobustCommand{\} }{\ifmmode\rbrace\else\textbraceright\fi}
259 \DeclareRobustCommand{\P }{\ifmmode\mathparagraph\else\textparagraph\fi}
260 \DeclareRobustCommand{\S }{\ifmmode\mathsection\else\textsection\fi}
261 \DeclareRobustCommand{\dag}{\ifmmode{\dagger}\else\textdagger\fi}
262 \DeclareRobustCommand{\ddag}{\ifmmode{\ddagger}\else\textdaggerdbl\fi}

```

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

```

263 \DeclareRobustCommand{\_ }{%
264   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
265 \DeclareRobustCommand{\copyright }{%
266   \ifmmode{\nfss@text{\textcopyright}}\else\textcopyright\fi}
267 \DeclareRobustCommand{\pounds }{%
268   \ifmmode\mathsterling\else\textsterling\fi}

269 \DeclareRobustCommand{\dots }{%
270   \ifmmode\mathellipsis\else\textellipsis\fi}

271 \let\ldots\dots

```

Default definition of comma below.

```

272 </2ekernel>
273 <latexrelease>\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%
274 <*2ekernel | latexrelease>
275 \DeclareTextCommandDefault\textcommabelow[1]
276   {\hmode@bgroup\ooalign{\null#1\crr\hidewidth\raise-.31ex
277   \hbox{\check@mathfonts\fontsize\ssf@size\z@
278   \math@fontsfalse\selectfont,}\hidewidth}\egroup}
279 <latexrelease>\EndIncludeInRelease
280 </2ekernel | latexrelease>

```

```

281 (latexrelease)\IncludeInRelease{0000/00/00}{\textcommabelow}{comma accent}%
282 (latexrelease)\let\textcommabelow\@undefined
283 (latexrelease)\EndIncludeInRelease

    Default definition of comma above (E.G.).
284 (latexrelease)\IncludeInRelease{2016/02/01}{\textcommaabove}{comma above}%
285 (*2ekernel | latexrelease)
286 \DeclareTextCommandDefault\textcommaabove[1]{%
287   \hmode@bgroup
288   \oalign{%
289     \hidewidth
290     \raise.7ex\hbox{%
291       \check@mathfonts\fontsize\ssf@size\z@\math@fontsfalse\selectfont‘%
292     }%
293     \hidewidth\crrc
294     \null#1\crrc
295   }%
296   \egroup
297 }
298 (latexrelease)\EndIncludeInRelease
299 (/2ekernel | latexrelease)
300 (latexrelease)\IncludeInRelease{0000/00/00}{\textcommaabove}{comma above}%
301 (latexrelease)\let\textcommaabove\@undefined
302 (latexrelease)\EndIncludeInRelease

```

19.5 Definitions for the OT1 encoding

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

Declare the encoding.

```

303 (*OT1)
304 \DeclareFontEncoding{OT1}{-}{-}

```

Declare the accents.

```

305 \DeclareTextAccent{"}{OT1}{127}
306 \DeclareTextAccent{'}{OT1}{19}
307 \DeclareTextAccent{.}{OT1}{95}
308 \DeclareTextAccent{=}{OT1}{22}
309 \DeclareTextAccent{\~}{OT1}{94}
310 \DeclareTextAccent{\`}{OT1}{18}
311 \DeclareTextAccent{\^}{OT1}{126}
312 \DeclareTextAccent{\H}{OT1}{125}
313 \DeclareTextAccent{\u}{OT1}{21}
314 \DeclareTextAccent{\v}{OT1}{20}
315 \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.

```

316 \DeclareTextCommand{\b}{OT1}[1]
317   {\hmode@bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
318     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
319 \DeclareTextCommand{\c}{OT1}[1]
320   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
321     \else\oalign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}\fi}

```

```

322 \DeclareTextCommand{\d}{OT1}[1]
323   {\hmode@bgroup
324     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}

```

Declare the text symbols.

```

325 \DeclareTextSymbol{\AE}{OT1}{29}
326 \DeclareTextSymbol{\OE}{OT1}{30}
327 \DeclareTextSymbol{\O}{OT1}{31}
328 \DeclareTextSymbol{\ae}{OT1}{26}
329 \DeclareTextSymbol{\i}{OT1}{16}
330 \DeclareTextSymbol{\j}{OT1}{17}
331 \DeclareTextSymbol{\oe}{OT1}{27}
332 \DeclareTextSymbol{\o}{OT1}{28}
333 \DeclareTextSymbol{\ss}{OT1}{25}
334 \DeclareTextSymbol{\textendash}{OT1}{124}
335 \DeclareTextSymbol{\textendash}{OT1}{123}

```

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

```

336 %\DeclareTextSymbol{\textexclamdown}{OT1}{60}
337 %\DeclareTextSymbol{\textquestiondown}{OT1}{62}
338 \DeclareTextCommand{\textexclamdown}{OT1}{!'}
339 \DeclareTextCommand{\textquestiondown}{OT1}{?'}
340 %\DeclareTextSymbol{\textthyphenchar}{OT1}{'\-}
341 %\DeclareTextSymbol{\textthyphen}{OT1}{'\-}
342 \DeclareTextSymbol{\textquotedblleft}{OT1}{92}
343 \DeclareTextSymbol{\textquotedblright}{OT1}{'\"}
344 \DeclareTextSymbol{\textquoteleft}{OT1}{'\'}
345 \DeclareTextSymbol{\textquoteright}{OT1}{'\'}

```

Some symbols which are faked from others:

```

346 % \DeclareTextCommand{\aa}{OT1}
347 %   {\accent23a}
348 \DeclareTextCommand{\L}{OT1}
349   {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
350 \DeclareTextCommand{\l}{OT1}
351   {\hmode@bgroup\@xxxii l\egroup}
352 % \DeclareTextCommand{\AA}{OT1}
353 %   {\leavevmode\setbox\z@\hbox{h}\dimen@ht\z@\advance\dimen@-1ex%
354 %     \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`.

```

355 \DeclareTextCompositeCommand{\r}{OT1}{A}
356   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
357     \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.

```

358 \DeclareTextCommand{\ij}{OT1}{%
359   \nobreak\hskip\z@skip i\kern-0.02em j\nobreak\hskip\z@skip}
360 \DeclareTextCommand{\IJ}{OT1}{%
361   \nobreak\hskip\z@skip I\kern-0.02em J\nobreak\hskip\z@skip}

```

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

```

362 \DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup

```

```

363 \ifdim \fontdimen\@ne\font >\z@
364     \slshape
365 \else
366     \upshape
367 \fi
368 \char'\$\egroup}

369 \DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
370 \ifdim \fontdimen\@ne\font >\z@
371     \itshape
372 \else
373     \fontshape{ui}\selectfont
374 \fi
375 \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.

```

376 \DeclareTextComposite{\.}{OT1}{i}{'\i}
377 \DeclareTextComposite{\.}{OT1}{\i}{'\i}
378 \DeclareTextCompositeCommand{\' }{OT1}{i}{\@tabacckludge'\i}
379 \DeclareTextCompositeCommand{\' }{OT1}{\i}{\@tabacckludge'\i}
380 \DeclareTextCompositeCommand{\^ }{OT1}{i}{\^i}
381 \DeclareTextCompositeCommand{\^ }{OT1}{\i}{\^i}

```

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

```

382 \DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
383 </OT1>

```

19.6 Definitions for the T1 encoding

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

Declare the encoding.

```

384 <*T1>
385 \DeclareFontEncoding{T1}{}{}

```

Declare the accents.

```

386 \DeclareTextAccent{\' }{T1}{0}
387 \DeclareTextAccent{\' }{T1}{1}
388 \DeclareTextAccent{\^ }{T1}{2}
389 \DeclareTextAccent{\^ }{T1}{3}
390 \DeclareTextAccent{\^ }{T1}{4}
391 \DeclareTextAccent{\H}{T1}{5}
392 \DeclareTextAccent{\r}{T1}{6}
393 \DeclareTextAccent{\v}{T1}{7}
394 \DeclareTextAccent{\u}{T1}{8}
395 \DeclareTextAccent{\=} {T1}{9}
396 \DeclareTextAccent{\. }{T1}{10}

```

Some accents have to be built by hand. Note that `\oalign` and `\o@lign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from `plain.tex` since that now has two incompatible definitions.

```

397 \DeclareTextCommand{\b}{T1}[1
398   {\hmode\bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
399     \vbox to.2ex{\hbox{\char9}\vss}\hidewidth}\egroup}
400 \DeclareTextCommand{\c}{T1}[1
401   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent11 #1%
402     \else{\oalign{\unhbox\z@\crrc
403       \hidewidth\char11\hidewidth}}\fi}
404 \DeclareTextCommand{\d}{T1}[1
405   {\hmode\bgroup
406     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
407 \DeclareTextCommand{\k}{T1}[1
408   {\hmode\bgroup\oalign{\null#1\crrc\hidewidth\char12}\egroup}
409 \DeclareTextCommand{\textogonekcentered}{T1}[1
410   {\hmode\bgroup\oalign{%
411     \null#1\crrc\hidewidth\char12\hidewidth}\egroup}

```

Some symbols are constructed.

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

```

412 \DeclareTextCommand{\textperthousand}{T1}
413   {\%\char 24 } % space or 'relax as delimiter?
414 \DeclareTextCommand{\textpertenthousand}{T1}
415   {\%\char 24\char 24 } % space or 'relax as delimiter?

```

Declare the text symbols.

```

416 %\DeclareTextSymbol{\AA}{T1}{197}
417 \DeclareTextSymbol{\AE}{T1}{198}
418 \DeclareTextSymbol{\DH}{T1}{208}
419 \DeclareTextSymbol{\DJ}{T1}{208}
420 \DeclareTextSymbol{\L}{T1}{138}
421 \DeclareTextSymbol{\NG}{T1}{141}
422 \DeclareTextSymbol{\OE}{T1}{215}
423 \DeclareTextSymbol{\O}{T1}{216}
424 \DeclareTextSymbol{\SS}{T1}{223}
425 \DeclareTextSymbol{\TH}{T1}{222}
426 %\DeclareTextSymbol{\aa}{T1}{229}
427 \DeclareTextSymbol{\ae}{T1}{230}
428 \DeclareTextSymbol{\dh}{T1}{240}
429 \DeclareTextSymbol{\dj}{T1}{158}
430 \DeclareTextSymbol{\guillemotleft}{T1}{19}
431 \DeclareTextSymbol{\guillemotright}{T1}{20}
432 \DeclareTextSymbol{\guilsinglleft}{T1}{14}
433 \DeclareTextSymbol{\guilsinglright}{T1}{15}
434 \DeclareTextSymbol{\i}{T1}{25}
435 \DeclareTextSymbol{\j}{T1}{26}
436 \DeclareTextSymbol{\ij}{T1}{188}
437 \DeclareTextSymbol{\IJ}{T1}{156}
438 \DeclareTextSymbol{\l}{T1}{170}
439 \DeclareTextSymbol{\ng}{T1}{173}
440 \DeclareTextSymbol{\oe}{T1}{247}
441 \DeclareTextSymbol{\o}{T1}{248}
442 \DeclareTextSymbol{\quotedblbase}{T1}{18}
443 \DeclareTextSymbol{\quotesinglbase}{T1}{13}
444 \DeclareTextSymbol{\ss}{T1}{255}
445 \DeclareTextSymbol{\textasciicircum}{T1}{'\^}

```

```

446 \DeclareTextSymbol{\textasciitilde}{T1}{'\~}
447 \DeclareTextSymbol{\textbackslash}{T1}{'\}
448 \DeclareTextSymbol{\textbar}{T1}{'|}
449 \DeclareTextSymbol{\textbraceleft}{T1}{'\{ }
450 \DeclareTextSymbol{\textbraceright}{T1}{'\} }
451 \DeclareTextSymbol{\textcompwordmark}{T1}{23}
452 \DeclareTextSymbol{\textdollar}{T1}{'\$}
453 \DeclareTextSymbol{\textemdash}{T1}{22}
454 \DeclareTextSymbol{\textendash}{T1}{21}
455 \DeclareTextSymbol{\textexclamdown}{T1}{189}
456 \DeclareTextSymbol{\textgreater}{T1}{'\>}
457 %\DeclareTextSymbol{\textthyphenchar}{T1}{127}
458 %\DeclareTextSymbol{\textthyphen}{T1}{'\-}
459 \DeclareTextSymbol{\textless}{T1}{'\<}
460 \DeclareTextSymbol{\textquestiondown}{T1}{190}
461 \DeclareTextSymbol{\textquotedblleft}{T1}{16}
462 \DeclareTextSymbol{\textquotedblright}{T1}{17}
463 \DeclareTextSymbol{\textquotedbl}{T1}{'\"}
464 \DeclareTextSymbol{\textquoteleft}{T1}{'\'}
465 \DeclareTextSymbol{\textquoteright}{T1}{'\'}
466 \DeclareTextSymbol{\textsection}{T1}{159}
467 \DeclareTextSymbol{\textsterling}{T1}{191}
468 \DeclareTextSymbol{\textunderscore}{T1}{95}
469 \DeclareTextSymbol{\textvisiblespace}{T1}{32}
470 \DeclareTextSymbol{\th}{T1}{254}

```

Declare the composites.

```

471 \DeclareTextComposite{\.}{T1}{i}{'\i}
472 \DeclareTextComposite{\.}{T1}{i}{'\i}
"80 = 128
473 \DeclareTextComposite{\u}{T1}{A}{128}
474 \DeclareTextComposite{\k}{T1}{A}{129}
475 \DeclareTextComposite{\'}{T1}{C}{130}
476 \DeclareTextComposite{\v}{T1}{C}{131}
477 \DeclareTextComposite{\v}{T1}{D}{132}
478 \DeclareTextComposite{\v}{T1}{E}{133}
479 \DeclareTextComposite{\k}{T1}{E}{134}
480 \DeclareTextComposite{\u}{T1}{G}{135}
"88 = 136
481 \DeclareTextComposite{\'}{T1}{L}{136}
482 \DeclareTextComposite{\v}{T1}{L}{137}
483 \DeclareTextComposite{\'}{T1}{N}{139}
484 \DeclareTextComposite{\v}{T1}{N}{140}
485 \DeclareTextComposite{\H}{T1}{O}{142}
486 \DeclareTextComposite{\'}{T1}{R}{143}
"90 = 144
487 \DeclareTextComposite{\v}{T1}{R}{144}
488 \DeclareTextComposite{\'}{T1}{S}{145}
489 \DeclareTextComposite{\v}{T1}{S}{146}
490 \DeclareTextComposite{\c}{T1}{S}{147}
491 \DeclareTextComposite{\v}{T1}{T}{148}
492 \DeclareTextComposite{\c}{T1}{T}{149}
493 \DeclareTextComposite{\H}{T1}{U}{150}

```

```

494 \DeclareTextComposite{\r}{T1}{U}{151}
"98 = 152
495 \DeclareTextComposite{"}{T1}{Y}{152}
496 \DeclareTextComposite{\'}{T1}{Z}{153}
497 \DeclareTextComposite{\v}{T1}{Z}{154}
498 \DeclareTextComposite{\.}{T1}{Z}{155}
499 \DeclareTextComposite{\.}{T1}{I}{157}
"A0 = 160
500 \DeclareTextComposite{\u}{T1}{a}{160}
501 \DeclareTextComposite{\k}{T1}{a}{161}
502 \DeclareTextComposite{\'}{T1}{c}{162}
503 \DeclareTextComposite{\v}{T1}{c}{163}
504 \DeclareTextComposite{\v}{T1}{d}{164}
505 \DeclareTextComposite{\v}{T1}{e}{165}
506 \DeclareTextComposite{\k}{T1}{e}{166}
507 \DeclareTextComposite{\u}{T1}{g}{167}
"A8 = 168
508 \DeclareTextComposite{\'}{T1}{l}{168}
509 \DeclareTextComposite{\v}{T1}{l}{169}
510 \DeclareTextComposite{\'}{T1}{n}{171}
511 \DeclareTextComposite{\v}{T1}{n}{172}
512 \DeclareTextComposite{\H}{T1}{o}{174}
513 \DeclareTextComposite{\'}{T1}{r}{175}
"B0 = 176
514 \DeclareTextComposite{\v}{T1}{r}{176}
515 \DeclareTextComposite{\'}{T1}{s}{177}
516 \DeclareTextComposite{\v}{T1}{s}{178}
517 \DeclareTextComposite{\c}{T1}{s}{179}
518 \DeclareTextComposite{\v}{T1}{t}{180}
519 \DeclareTextComposite{\c}{T1}{t}{181}
520 \DeclareTextComposite{\H}{T1}{u}{182}
521 \DeclareTextComposite{\r}{T1}{u}{183}
"B8 = 184
522 \DeclareTextComposite{"}{T1}{y}{184}
523 \DeclareTextComposite{\'}{T1}{z}{185}
524 \DeclareTextComposite{\v}{T1}{z}{186}
525 \DeclareTextComposite{\.}{T1}{z}{187}
"C0 = 192
526 \DeclareTextComposite{\'}{T1}{A}{192}
527 \DeclareTextComposite{\'}{T1}{A}{193}
528 \DeclareTextComposite{\^}{T1}{A}{194}
529 \DeclareTextComposite{\~}{T1}{A}{195}
530 \DeclareTextComposite{"}{T1}{A}{196}
531 \DeclareTextComposite{\r}{T1}{A}{197}
532 \DeclareTextComposite{\c}{T1}{C}{199}
"C8 = 200
533 \DeclareTextComposite{\'}{T1}{E}{200}
534 \DeclareTextComposite{\'}{T1}{E}{201}
535 \DeclareTextComposite{\~}{T1}{E}{202}
536 \DeclareTextComposite{"}{T1}{E}{203}

```



```

537 \DeclareTextComposite{\'}{T1}{I}{204}
538 \DeclareTextComposite{\'}{T1}{I}{205}
539 \DeclareTextComposite{\^}{T1}{I}{206}
540 \DeclareTextComposite{\"}{T1}{I}{207}
"D0 = 208
541 \DeclareTextComposite{\~}{T1}{N}{209}
542 \DeclareTextComposite{\'}{T1}{O}{210}
543 \DeclareTextComposite{\'}{T1}{O}{211}
544 \DeclareTextComposite{\^}{T1}{O}{212}
545 \DeclareTextComposite{\~}{T1}{O}{213}
546 \DeclareTextComposite{\"}{T1}{O}{214}
"D8 = 216
547 \DeclareTextComposite{\'}{T1}{U}{217}
548 \DeclareTextComposite{\'}{T1}{U}{218}
549 \DeclareTextComposite{\^}{T1}{U}{219}
550 \DeclareTextComposite{\"}{T1}{U}{220}
551 \DeclareTextComposite{\'}{T1}{Y}{221}
"E0 = 224
552 \DeclareTextComposite{\'}{T1}{a}{224}
553 \DeclareTextComposite{\'}{T1}{a}{225}
554 \DeclareTextComposite{\^}{T1}{a}{226}
555 \DeclareTextComposite{\~}{T1}{a}{227}
556 \DeclareTextComposite{\"}{T1}{a}{228}
557 \DeclareTextComposite{\r}{T1}{a}{229}
558 \DeclareTextComposite{\c}{T1}{c}{231}
"E8 = 232
559 \DeclareTextComposite{\'}{T1}{e}{232}
560 \DeclareTextComposite{\'}{T1}{e}{233}
561 \DeclareTextComposite{\^}{T1}{e}{234}
562 \DeclareTextComposite{\"}{T1}{e}{235}
563 \DeclareTextComposite{\'}{T1}{i}{236}
564 \DeclareTextComposite{\'}{T1}{i}{236}
565 \DeclareTextComposite{\'}{T1}{i}{237}
566 \DeclareTextComposite{\'}{T1}{i}{237}
567 \DeclareTextComposite{\^}{T1}{i}{238}
568 \DeclareTextComposite{\^}{T1}{i}{238}
569 \DeclareTextComposite{\"}{T1}{i}{239}
570 \DeclareTextComposite{\"}{T1}{i}{239}
"F0 = 240
571 \DeclareTextComposite{\~}{T1}{n}{241}
572 \DeclareTextComposite{\'}{T1}{o}{242}
573 \DeclareTextComposite{\'}{T1}{o}{243}
574 \DeclareTextComposite{\^}{T1}{o}{244}
575 \DeclareTextComposite{\~}{T1}{o}{245}
576 \DeclareTextComposite{\"}{T1}{o}{246}
"F8 = 248
577 \DeclareTextComposite{\'}{T1}{u}{249}
578 \DeclareTextComposite{\'}{T1}{u}{250}
579 \DeclareTextComposite{\^}{T1}{u}{251}
580 \DeclareTextComposite{\"}{T1}{u}{252}
581 \DeclareTextComposite{\'}{T1}{y}{253}

```

```

582 \DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
583 \DeclareTextCompositeCommand{\k}{T1}{0}{\textogonekcentered{0}}

584 \DeclareTextCompositeCommand{\c}{T1}{G}{\textcommabelow{G}}
585 \DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
586 \DeclareTextCompositeCommand{\c}{T1}{K}{\textcommabelow{K}}
587 \DeclareTextCompositeCommand{\c}{T1}{k}{\textcommabelow{k}}
588 \DeclareTextCompositeCommand{\c}{T1}{L}{\textcommabelow{L}}
589 \DeclareTextCompositeCommand{\c}{T1}{l}{\textcommabelow{l}}
590 \DeclareTextCompositeCommand{\c}{T1}{N}{\textcommabelow{N}}
591 \DeclareTextCompositeCommand{\c}{T1}{n}{\textcommabelow{n}}
592 \DeclareTextCompositeCommand{\c}{T1}{R}{\textcommabelow{R}}
593 \DeclareTextCompositeCommand{\c}{T1}{r}{\textcommabelow{r}}
594 </T1>

```

19.7 Definitions for the OMS encoding

The definitions for the ‘ \TeX math symbol’ (OMS) encoding. Even though this is meant to be a math font, it includes some of the standard \LaTeX text symbols.

Declare the encoding.

```

595 (*OMS)
596 \DeclareFontEncoding{OMS}{}{}

```

Declare the symbols.

```

597 % \changes{v1.99}{2004/02/02}{Added \cs{textbigcircle}}
598 %   Note that slot 13 has in places been named |\Orb|: please root
599 %   out and destroy this impolity wherever you find it!
600 %   \begin{macrocode}
601 \DeclareTextSymbol{\textasteriskcentered}{OMS}{3}    % "03
602 \DeclareTextSymbol{\textbackslash}{OMS}{110}        % "6E
603 \DeclareTextSymbol{\textbar}{OMS}{106}              % "6A
604 \DeclareTextSymbol{\textbardbl}{OMS}{107}          % "6B
605 \DeclareTextSymbol{\textbraceleft}{OMS}{102}       % "66
606 \DeclareTextSymbol{\textbraceright}{OMS}{103}      % "67
607 \DeclareTextSymbol{\textbullet}{OMS}{15}           % "0F
608 \DeclareTextSymbol{\textdaggerdbl}{OMS}{122}       % "7A
609 \DeclareTextSymbol{\textdagger}{OMS}{121}          % "79
610 \DeclareTextSymbol{\textparagraph}{OMS}{123}       % "7B
611 \DeclareTextSymbol{\textperiodcentered}{OMS}{1}    % "01
612 \DeclareTextSymbol{\textsection}{OMS}{120}         % "78
613 \DeclareTextSymbol{\textbigcircle}{OMS}{13}       % "0D
614 \DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
615   \oalign{%
616     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
617     \char 13 % "0D
618   }%
619 \egroup}
620 </OMS>

```

19.8 Definitions for the OML encoding

The definitions for the ‘ \TeX math italic’ (OML) encoding. Even though this is meant to be a math font, it includes some of the standard \LaTeX text symbols.

Declare the encoding.

```
621 (*OML)
622 \DeclareFontEncoding{OML}{-}{}
```

Declare the symbols.

```
623 \DeclareTextSymbol{\textless}{OML}{'\<}
624 \DeclareTextSymbol{\textgreater}{OML}{'\>}
625 \DeclareTextAccent{\t}{OML}{127} % "7F
626 \end{OML}
```

19.9 Definitions for the OT4 encoding

These definitions are for the Polish extension to the ‘ \TeX text’ (OT1) encoding. This encoding was created by B. Jackowski and M. Ryćko for use with the Polish version of Computer Modern and Computer Concrete. In positions 0–127 it is identical to OT1 but it contains some additional characters in the upper half. The \LaTeX support was developed by Mariusz Olko.

The PL fonts that use it are available as follows:

Metafont sources <ftp://ftp.gust.org.pl/TeX/language/polish/pl-mf.zip>;

Font files <ftp://ftp.gust.org.pl/TeX/language/polish/pl-tfm.zip>.

Declare the encoding.

```
627 (*OT4)
628 \DeclareFontEncoding{OT4}{-}{-}
629 \DeclareFontSubstitution{OT4}{cmr}{m}{n}
```

Declare the accents.

```
630 \DeclareTextAccent{"}{OT4}{127}
631 \DeclareTextAccent{'}{OT4}{19}
632 \DeclareTextAccent{\.}{OT4}{95}
633 \DeclareTextAccent{=} {OT4}{22}
634 \DeclareTextAccent{^}{OT4}{94}
635 \DeclareTextAccent{\`}{OT4}{18}
636 \DeclareTextAccent{\~}{OT4}{126}
637 \DeclareTextAccent{\H}{OT4}{125}
638 \DeclareTextAccent{\u}{OT4}{21}
639 \DeclareTextAccent{\v}{OT4}{20}
640 \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:

```
641 \DeclareTextCommand{\k}{OT4}[1]{%
642   \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.

```
643 \DeclareTextCommand{\b}{OT4}[1]
644   {\hmode@bgroup\o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-3ex}%
645     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
646 \DeclareTextCommand{\c}{OT4}[1]
647   {\leavevmode\setbox\z@hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
648     \else{\oalign{\unhbox\z@\crrc\hidewidth\char24\hidewidth}}\fi}
649 \DeclareTextCommand{\d}{OT4}[1]
650   {\hmode@bgroup
651     \o@lign{\relax#1\crrc\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
```

Declare the text symbols.

```
652 \DeclareTextSymbol{\AE}{OT4}{29}
653 \DeclareTextSymbol{\OE}{OT4}{30}
654 \DeclareTextSymbol{\O}{OT4}{31}
655 \DeclareTextSymbol{\L}{OT4}{138}
656 \DeclareTextSymbol{\ae}{OT4}{26}
657 \DeclareTextSymbol{\guillemotleft}{OT4}{174}
658 \DeclareTextSymbol{\guillemotright}{OT4}{175}
659 \DeclareTextSymbol{\i}{OT4}{16}
660 \DeclareTextSymbol{\j}{OT4}{17}
661 \DeclareTextSymbol{\l}{OT4}{170}
662 \DeclareTextSymbol{\o}{OT4}{28}
663 \DeclareTextSymbol{\oe}{OT4}{27}
664 \DeclareTextSymbol{\quotedblbase}{OT4}{255}
665 \DeclareTextSymbol{\ss}{OT4}{25}
666 \DeclareTextSymbol{\textendash}{OT4}{124}
667 \DeclareTextSymbol{\textendash}{OT4}{123}
668 \DeclareTextSymbol{\textexclamdown}{OT4}{60}
669 %\DeclareTextSymbol{\texthyphenchar}{OT4}{'\-}
670 %\DeclareTextSymbol{\texthyphen}{OT4}{'\-}
671 \DeclareTextSymbol{\textquestiondown}{OT4}{62}
672 \DeclareTextSymbol{\textquotedblleft}{OT4}{92}
673 \DeclareTextSymbol{\textquotedblright}{OT4}{'" }
674 \DeclareTextSymbol{\textquoteleft}{OT4}{'\'}
675 \DeclareTextSymbol{\textquoteright}{OT4}{'\'}
```

Definition for Å as in OT1:

```
676 \DeclareTextCompositeCommand{\r}{OT4}{A}
677   {\leavevmode\setbox\z@\hbox{!}\dimen@ht\z@\advance\dimen@-1ex%
678   \rlap{\raise.67\dimen@\hbox{\char23}}A}
```

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

```
679 \DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
680   \ifdim \fontdimen\@ne\font >\z@
681     \slshape
682   \else
683     \upshape
684   \fi
685   \char'\$\egroup}
686 \DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
687   \ifdim \fontdimen\@ne\font >\z@
688     \itshape
689   \else
690     \fontshape{ui}\selectfont
691   \fi
692   \char'\$\egroup}
```

Declare the composites.

```
693 \DeclareTextComposite{\k}{OT4}{A}{129}
694 \DeclareTextComposite{\'}{OT4}{C}{130}
695 \DeclareTextComposite{\k}{OT4}{E}{134}
696 \DeclareTextComposite{\'}{OT4}{N}{139}
697 \DeclareTextComposite{\'}{OT4}{S}{145}
698 \DeclareTextComposite{\'}{OT4}{Z}{153}
699 \DeclareTextComposite{\.}{OT4}{Z}{155}
```

```

700 \DeclareTextComposite{\k}{OT4}{a}{161}
701 \DeclareTextComposite{\'}{OT4}{c}{162}
702 \DeclareTextComposite{\k}{OT4}{e}{166}
703 \DeclareTextComposite{\'}{OT4}{n}{171}
704 \DeclareTextComposite{\'}{OT4}{s}{177}
705 \DeclareTextComposite{\'}{OT4}{z}{185}
706 \DeclareTextComposite{\.}{OT4}{z}{187}
707 \DeclareTextComposite{\'}{OT4}{0}{211}
708 \DeclareTextComposite{\'}{OT4}{o}{243}
709 /OT4

```

19.10 Definitions for the TS1 encoding

```

710 (*TS1)
711 \DeclareFontEncoding{TS1}{}{}
712 \DeclareFontSubstitution{TS1}{cmr}{m}{n}

```

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

```

713 \DeclareTextCommand{\capitalcedilla}{TS1}[1]
714   {\hmode@bgroup
715    \oalign{\null#1\cr\cr\hidewidth\char11\hidewidth}\egroup}
716 \DeclareTextCommand{\capitalogonek}{TS1}[1]
717   {\hmode@bgroup
718    \oalign{\null#1\cr\cr\hidewidth\char12\hidewidth}\egroup}

```

Accents for capital letters.

These commands can be used by the end user either directly or through definitions of the type

```
\DeclareTextCompositeCommand{\'}{T1}{X}{\capitalacute X}
```

None of the latter definitions are provided by default, since they are probably rarely used.

”00 = 0

```

719 \DeclareTextAccent{\capitalgrave}{TS1}{0}
720 \DeclareTextAccent{\capitalacute}{TS1}{1}
721 \DeclareTextAccent{\capitalcircumflex}{TS1}{2}
722 \DeclareTextAccent{\capitaltilde}{TS1}{3}
723 \DeclareTextAccent{\capitaldieresis}{TS1}{4}
724 \DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}
725 \DeclareTextAccent{\capitalring}{TS1}{6}
726 \DeclareTextAccent{\capitalcaron}{TS1}{7}

```

”08 = 8

```

727 \DeclareTextAccent{\capitalbreve}{TS1}{8}
728 \DeclareTextAccent{\capitalmacron}{TS1}{9}
729 \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.

” =

```

730 \DeclareTextAccent{\t}{TS1}{26}
731 \DeclareTextAccent{\capitaltie}{TS1}{27}
732 \DeclareTextAccent{\newtie}{TS1}{28}
733 \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`.

```

734 \DeclareTextSymbol{\textcapitalcompwordmark}{TS1}{23}
735 \DeclareTextSymbol{\textascendercompwordmark}{TS1}{31}

```

The text companion symbols.

```

736 \DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
"10 = 16
737 \DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
738 \DeclareTextSymbol{\texttwelveudash}{TS1}{21}
739 \DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
"18 = 24
740 \DeclareTextSymbol{\textleftarrow}{TS1}{24}
741 \DeclareTextSymbol{\textrightarrow}{TS1}{25}
"20 = 32
742 \DeclareTextSymbol{\textblank}{TS1}{32}
743 \DeclareTextSymbol{\textdollar}{TS1}{36}
744 \DeclareTextSymbol{\textquotesingle}{TS1}{39}
"28 = 40
745 \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.

```

746 \DeclareTextSymbol{\textdblhyphen}{TS1}{45}
747 \DeclareTextSymbol{\textfractionsolidus}{TS1}{47}

```

Oldstyle digits.

```

"30 = 48
748 \DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
749 \DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
750 \DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
751 \DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
752 \DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
753 \DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
754 \DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
755 \DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}
"38 = 56
756 \DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
757 \DeclareTextSymbol{\textnineoldstyle}{TS1}{57}

```

More text companion symbols.

```

758 \DeclareTextSymbol{\textlangle}{TS1}{60}
759 \DeclareTextSymbol{\textminus}{TS1}{61}
760 \DeclareTextSymbol{\textrightangle}{TS1}{62}
"48 = 72
761 \DeclareTextSymbol{\textmho}{TS1}{77}

```

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

```
762 \DeclareTextSymbol{\textbigcircle}{TS1}{79}
763 \DeclareTextCommand{\textcircled}{TS1}[1]{\hmode@bgroup
764   \ooalign{%
765     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
766     \char 79   % '117 = "4F
767   }%
768 \egroup}
```

More text companion symbols.

"50 = 80

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

"58 = 88

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

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

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

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

"60 = 96

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

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

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

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

"68 = 104

```
778 \DeclareTextSymbol{\textleaf}{TS1}{108}
```

```
779 \DeclareTextSymbol{\textmarried}{TS1}{109}
```

```
780 \DeclareTextSymbol{\textmusicalnote}{TS1}{110}
```

"78 = 120

```
781 \DeclareTextSymbol{\texttildelow}{TS1}{126}
```

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

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

"80 = 128

```
783 \DeclareTextSymbol{\textasciibreve}{TS1}{128}
```

```
784 \DeclareTextSymbol{\textasciicaron}{TS1}{129}
```

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

```
785 \DeclareTextSymbol{\textacutedbl}{TS1}{130}
```

```
786 \DeclareTextSymbol{\textgravedbl}{TS1}{131}
```

```
787 \DeclareTextSymbol{\textdagger}{TS1}{132}
```

```
788 \DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
```

```
789 \DeclareTextSymbol{\textbardbl}{TS1}{134}
```

```
790 \DeclareTextSymbol{\textperthousand}{TS1}{135}
```

"88 = 136

```
791 \DeclareTextSymbol{\textbullet}{TS1}{136}
```

```
792 \DeclareTextSymbol{\textcelsius}{TS1}{137}
```

```
793 \DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
```

```
794 \DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
```

```
795 \DeclareTextSymbol{\textflorin}{TS1}{140}
```

```

796 \DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
797 \DeclareTextSymbol{\textwon}{TS1}{142}
798 \DeclareTextSymbol{\textnaira}{TS1}{143}
"90 = 144
799 \DeclareTextSymbol{\textguarani}{TS1}{144}
800 \DeclareTextSymbol{\textpeso}{TS1}{145}
801 \DeclareTextSymbol{\textlira}{TS1}{146}
802 \DeclareTextSymbol{\textrecipe}{TS1}{147}
803 \DeclareTextSymbol{\textinterrobang}{TS1}{148}
804 \DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
805 \DeclareTextSymbol{\textdong}{TS1}{150}
806 \DeclareTextSymbol{\texttrademark}{TS1}{151}

```

"98 = 152

```

807 \DeclareTextSymbol{\textpertenthousand}{TS1}{152}
808 \DeclareTextSymbol{\textpilcrow}{TS1}{153}
809 \DeclareTextSymbol{\textbaht}{TS1}{154}
810 \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`.

```

811 \DeclareTextSymbol{\textdiscount}{TS1}{156}
812 \DeclareTextSymbol{\textestimated}{TS1}{157}
813 \DeclareTextSymbol{\textopenbullet}{TS1}{158}
814 \DeclareTextSymbol{\textservicemark}{TS1}{159}

```

"A0 = 160

```

815 \DeclareTextSymbol{\textlquill}{TS1}{160}
816 \DeclareTextSymbol{\textrquill}{TS1}{161}
817 \DeclareTextSymbol{\textcent}{TS1}{162}
818 \DeclareTextSymbol{\textsterling}{TS1}{163}
819 \DeclareTextSymbol{\textcurrency}{TS1}{164}
820 \DeclareTextSymbol{\textyen}{TS1}{165}
821 \DeclareTextSymbol{\textbrokenbar}{TS1}{166}
822 \DeclareTextSymbol{\textsection}{TS1}{167}

```

"A8 = 168

```

823 \DeclareTextSymbol{\textasciidieresis}{TS1}{168}
824 \DeclareTextSymbol{\textcopyright}{TS1}{169}
825 \DeclareTextSymbol{\textordfeminine}{TS1}{170}
826 \DeclareTextSymbol{\textcopyleft}{TS1}{171}
827 \DeclareTextSymbol{\textlnot}{TS1}{172}

```

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

```

828 \DeclareTextSymbol{\textcircledP}{TS1}{173}
829 \DeclareTextSymbol{\textregistered}{TS1}{174}
830 \DeclareTextSymbol{\textasciimacron}{TS1}{175}

```

"B0 = 176

```

831 \DeclareTextSymbol{\textdegree}{TS1}{176}
832 \DeclareTextSymbol{\textpm}{TS1}{177}
833 \DeclareTextSymbol{\texttwosuperior}{TS1}{178}
834 \DeclareTextSymbol{\textthreesuperior}{TS1}{179}
835 \DeclareTextSymbol{\textasciiacute}{TS1}{180}

```



```

836 \DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
837 \DeclareTextSymbol{\textparagraph}{TS1}{182}
838 \DeclareTextSymbol{\textperiodcentered}{TS1}{183}
" B8 = 184
839 \DeclareTextSymbol{\textreferencemark}{TS1}{184}
840 \DeclareTextSymbol{\textonesuperior}{TS1}{185}
841 \DeclareTextSymbol{\textordmasculine}{TS1}{186}
842 \DeclareTextSymbol{\textsurd}{TS1}{187}
843 \DeclareTextSymbol{\textonequarter}{TS1}{188}
844 \DeclareTextSymbol{\textonehalf}{TS1}{189}
845 \DeclareTextSymbol{\textthreequarters}{TS1}{190}
846 \DeclareTextSymbol{\texteuro}{TS1}{191}
" E0 = 208
847 \DeclareTextSymbol{\texttimes}{TS1}{214}
" F0 = 240
848 \DeclareTextSymbol{\textdiv}{TS1}{246}
849 </TS1>

```

20 Package files

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

20.1 The fontenc package

This package allows authors to specify which encodings they will use. For each encoding F00, the package looks to see if the encoding F00 has already been declared. If it has not, the file `fooenc.def` is loaded. The default encoding is set to be F00.

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.

```
850 (*package)
```

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

```

851 \def\update@uclc@with@cyrillic{%
852 \expandafter\def\expandafter\@uclclist\expandafter
853 {\@uclclist
854 \cyra\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
855 \CYRABHDZE\cyrabhha\CYRABHHA\cyrae\CYRAE\cyrb\CYRB\cyrbyus
856 \CYRBYUS\cyrc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrdsc
857 \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
858 \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
859 \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
860 \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
861 \cyrgchcrs\CYRGHCRS\cyrgkh\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
862 \cyrhds\CYRHDS\cyrhcrs\CYRHCRS\cyrhkh\CYRHKH\cyrhdsn

```

```

863 \CYRHRDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
864 \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
865 \cyrkbeak\CYRKBEAK\cyrkdsc\CYRKDSC\cyrkhcrs\CYRKHCRS\cyrkhk
866 \CYRKHK\cyrkvcrs\CYRKVCRS\cyr1\CYRL\cyrlldsc\CYRLDSC\cyr1hk
867 \CYRLHK\cyr1je\CYRLJE\cyrm\CYRM\cyrmdsc\CYRMDSC\cyrmhk\CYRMHK
868 \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnhk\CYRNHK\cyrnje
869 \CYRNJE\cyrnlhk\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
870 \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
871 \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
872 \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
873 \cyrsftsn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
874 \cyrt\CYRT\cyrtldsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
875 \cyruc\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyrz\CYRZ
876 \CYRYA\CYRYA\cyrz\CYRZ\cyrzhdsc\CYRZHDSC\cyrzh\CYRZH
877 \cyrzhdsc\CYRZHDSC}%
878 \let\update@uclc@with@cyrillic\relax
879 }
880 }

```

Here we process each option:

```

881 \DeclareOption*{%
882   \let\encodingdefault\CurrentOption
883   \edef\reserved@f{%
884     \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
885   \reserved@f
886   \InputIfFileExists\reserved@f
887     {}{\PackageError{fontenc}%
888       {Encoding file '\reserved@f' not found.%
889       \MessageBreak
890       You might have misspelt the name of the encoding}%
891     {Necessary code for this encoding was not
892     loaded.\MessageBreak
893     Thus calling the encoding later on will
894     produce further error messages.}}%
895   \let\reserved@f\relax

```

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

```

896 \expandafter\in@\expandafter{\CurrentOption}%
897   {T2A,T2B,T2C,X2,LCY,OT2}%
898 \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.

```

899   \expandafter\in@\expandafter\cyra\expandafter
900     {\@uclclist}%
901   \ifin@
902   \else
903     \update@uclc@with@cyrillic
904   \fi
905 \fi
906 }
907 \ProcessOptions*

```

```
908 \fontencoding\encodingdefault\selectfont
```

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

```
909 \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.

```
910 \global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
```

```
911 \global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
```

```
912 \global\let\@ifl@ter@@\@ifl@ter
```

```
913 \def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
```

```
914 \endpackage
```

20.2 The `textcomp` package

This one is for the `TS1` encoding which contains text symbols for use with the `T1`-encoded text fonts. It therefore first inputs the file `TS1enc.def` and then sets (or resets) the defaults for the symbols it contains. The result of this is that when one of these symbols is accessed and the current encoding does not provide it, the symbol will be supplied by a silent, local change to this encoding.

```
915 (*TS1sty)
```

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:

```
916 \PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
```

```
917   \space\space 5 = only ISO-Adobe without
```

```

918                                     \string\textcurrency\MessageBreak
919 \space\space 4 = 5 + \string\texteuro\MessageBreak
920 \space\space 3 = 4 + \string\textohm\MessageBreak
921 \space\space 2 = 3 + \noexpand\textestimated+
922                                     \string\textcurrency\MessageBreak
923 \space\space 1 = TS1 - \noexpand\textcircled-
924                                     \string\t\MessageBreak
925 \space\space 0 = TS1 (full)\MessageBreak
926 Font families with sub-encoding setting implement\MessageBreak
927 only a restricted character set as indicated.\MessageBreak
928 Family '?' is the default used for unknown fonts.\MessageBreak
929 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 ?.

```

930 \def\DeclareEncodingSubset#1#2#3{%
931   \ifundefined{#1:#2}%
932     {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
933     {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
934   \@namedef{#1:#2}{#3}}
935 \@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

```

936 \newif\iftc@forced   \tc@forcedfalse

```

This is implemented by defining the default subset:

```

937 \DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
938 \DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
939 \DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
940 \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.

```
941 \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.

```
942 \def\tc@errorwarn{\PackageError}
```

```
943 \DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
```

```
944 \ExecuteOptions{almostfull}
```

```
945 \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.

```
946 \iftc@forced
```

If the “force” option was given we always use the default for testing against.

```
947 \def\CheckEncodingSubset#1#2#3#4#5{%
```

```
948   \ifnum #4>%
```

```
949     0\csname #2:\endcsname
```

```
950     \relax
```

```
951   \expandafter\@firstoftwo
```

```
952   \else
```

```
953   \expandafter\@secondoftwo
```

```
954   \fi
```

```
955   {#1{#2}}{#3}%
```

```
956   #5%
```

```
957 }
```

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.

```
958 \else
```

```
959 \def\CheckEncodingSubset#1#2#3#4#5{%
```

```
960   \ifnum #4>%
```

```
961     \expandafter\ifx\csname #2:\f@family\endcsname\relax
```

```
962     0\csname #2:\endcsname
```

```
963     \else
```

```

964     \csname #2:\f@family\endcsname
965     \fi
966     \relax
967     \expandafter\@firstoftwo
968     \else
969     \expandafter\@secondoftwo
970     \fi
971     {#1{#2}}{#3}%
972     #5%
973 }
974 \fi

```

tc@subst

```

975 \def\tc@subst#1{%
976   \tc@errorwarn{textcomp}% % should be latex error if general
977   {Symbol \string#1 not provided by\MessageBreak
978     font family \f@family\space
979     in TS1 encoding.\MessageBreak Default family used instead}\@eha
980   \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
981 }

```

\textcompsubstdefault

```

982 \def\textcompsubstdefault{cmr}

```

\tc@error \tc@error is going to be used in arg #3 of \CheckEncodingSubset when a symbol is not available in a certain font family. It gets pass the encoding it normally lives in (arg one) and the name of the symbol or accent that has a problem.

```

983 % error commands take argument:
984 % #1 symbol to be used
985 \def\tc@error#1{%
986   \PackageError{textcomp}% % should be latex error if general
987   {Accent \string#1 not provided by\MessageBreak
988     font family \f@family\space
989     in TS1 encoding}\@eha
990 }

```

\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 “=”.

```

991 \def\tc@fake@euro#1{%
992   \leavevmode
993   \PackageInfo{textcomp}{Faking \noexpand#1for font family
994     \f@family\MessageBreak in TS1 encoding}%
995   \valign{##\cr
996     \vfil\hbox to 0.07em{\dimen@\f@size\p@
997       \math@fontsfalse
998       \fontsize{.7\dimen@}\z@\selectfont=\hss}%
999     \vfil\cr%
1000   \hbox{C}\crr
1001   }%
1002 }

```

`\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.

```
1003 \def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}  
1004 \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...

```
1005 \DeclareTextAccentDefault{\capitalcedilla}{TS1}  
1006 \DeclareTextAccentDefault{\capitalogonek}{TS1}  
1007 \DeclareTextAccentDefault{\capitalgrave}{TS1}  
1008 \DeclareTextAccentDefault{\capitalacute}{TS1}  
1009 \DeclareTextAccentDefault{\capitalcircumflex}{TS1}  
1010 \DeclareTextAccentDefault{\capitaltilde}{TS1}  
1011 \DeclareTextAccentDefault{\capitaldieresis}{TS1}  
1012 \DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}  
1013 \DeclareTextAccentDefault{\capitalring}{TS1}  
1014 \DeclareTextAccentDefault{\capitalcaron}{TS1}  
1015 \DeclareTextAccentDefault{\capitalbreve}{TS1}  
1016 \DeclareTextAccentDefault{\capitalmacron}{TS1}  
1017 \DeclareTextAccentDefault{\capitaldotaccent}{TS1}
```

... and then the other glyphs.

```
1018 \DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}  
1019 \DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}  
1020 \DeclareTextSymbolDefault{\textquotestraightbase}{TS1}  
1021 \DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}  
1022 \DeclareTextSymbolDefault{\texttwelveudash}{TS1}  
1023 \DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}  
1024 \DeclareTextSymbolDefault{\textdollar}{TS1}  
1025 \DeclareTextSymbolDefault{\textquotesingle}{TS1}  
1026 \DeclareTextSymbolDefault{\textasteriskcentered}{TS1}  
1027 \DeclareTextSymbolDefault{\textfractionsolidus}{TS1}  
1028 \DeclareTextSymbolDefault{\textminus}{TS1}  
1029 \DeclareTextSymbolDefault{\textlbrackdbl}{TS1}  
1030 \DeclareTextSymbolDefault{\textrbrackdbl}{TS1}  
1031 \DeclareTextSymbolDefault{\textasciigrave}{TS1}  
1032 \DeclareTextSymbolDefault{\texttildelow}{TS1}  
1033 \DeclareTextSymbolDefault{\textasciibreve}{TS1}  
1034 \DeclareTextSymbolDefault{\textasciicaron}{TS1}  
1035 \DeclareTextSymbolDefault{\textgravedbl}{TS1}  
1036 \DeclareTextSymbolDefault{\textacutedbl}{TS1}  
1037 \DeclareTextSymbolDefault{\textdagger}{TS1}  
1038 \DeclareTextSymbolDefault{\textdaggerdbl}{TS1}  
1039 \DeclareTextSymbolDefault{\textbardbl}{TS1}  
1040 \DeclareTextSymbolDefault{\textperthousand}{TS1}  
1041 \DeclareTextSymbolDefault{\textbullet}{TS1}  
1042 \DeclareTextSymbolDefault{\textcelsius}{TS1}  
1043 \DeclareTextSymbolDefault{\textflorin}{TS1}
```

```

1044 \DeclareTextSymbolDefault{\texttrademark}{TS1}
1045 \DeclareTextSymbolDefault{\textcent}{TS1}
1046 \DeclareTextSymbolDefault{\textsterling}{TS1}
1047 \DeclareTextSymbolDefault{\textyen}{TS1}
1048 \DeclareTextSymbolDefault{\textbrokenbar}{TS1}
1049 \DeclareTextSymbolDefault{\textsection}{TS1}
1050 \DeclareTextSymbolDefault{\textasciidieresis}{TS1}
1051 \DeclareTextSymbolDefault{\textcopyright}{TS1}
1052 \DeclareTextSymbolDefault{\textordfeminine}{TS1}
1053 \DeclareTextSymbolDefault{\textlnot}{TS1}
1054 \DeclareTextSymbolDefault{\textregistered}{TS1}
1055 \DeclareTextSymbolDefault{\textasciimacron}{TS1}
1056 \DeclareTextSymbolDefault{\textdegree}{TS1}
1057 \DeclareTextSymbolDefault{\textpm}{TS1}
1058 \DeclareTextSymbolDefault{\texttwosuperior}{TS1}
1059 \DeclareTextSymbolDefault{\textthreesuperior}{TS1}
1060 \DeclareTextSymbolDefault{\textasciacute}{TS1}
1061 \DeclareTextSymbolDefault{\textmu}{TS1}
1062 \DeclareTextSymbolDefault{\textparagraph}{TS1}
1063 \DeclareTextSymbolDefault{\textperiodcentered}{TS1}
1064 \DeclareTextSymbolDefault{\textonesuperior}{TS1}
1065 \DeclareTextSymbolDefault{\textordmasculine}{TS1}
1066 \DeclareTextSymbolDefault{\textonequarter}{TS1}
1067 \DeclareTextSymbolDefault{\textonehalf}{TS1}
1068 \DeclareTextSymbolDefault{\textthreequarters}{TS1}
1069 \DeclareTextSymbolDefault{\texttimes}{TS1}
1070 \DeclareTextSymbolDefault{\textdiv}{TS1}

```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```

1071 \DeclareTextCommandDefault{\texteuro}
1072   {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}

```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```

1073 \DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}

```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```

1074 \DeclareTextCommandDefault{\textestimated}%
1075   {\tc@check@symbol3\textestimated}
1076 \DeclareTextCommandDefault{\textcurrency}%
1077   {\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.

```

1078 \DeclareTextCommandDefault{\capitaltie}%
1079   {\tc@check@accent2\capitaltie}
1080 \DeclareTextCommandDefault{\newtie}%
1081   {\tc@check@accent2\newtie}
1082 \DeclareTextCommandDefault{\capitalnewtie}%
1083   {\tc@check@accent2\capitalnewtie}
1084 \DeclareTextCommandDefault{\textleftarrow}%
1085   {\tc@check@symbol2\textleftarrow}
1086 \DeclareTextCommandDefault{\textrightarrow}%

```



```

1087     {\tc@check@symbol2\extrightarrow}
1088 \DeclareTextCommandDefault{\textblank}%
1089     {\tc@check@symbol2\textblank}
1090 \DeclareTextCommandDefault{\textdblhyphen}%
1091     {\tc@check@symbol2\textdblhyphen}
1092 \DeclareTextCommandDefault{\textzerooldstyle}%
1093     {\tc@check@symbol2\textzerooldstyle}
1094 \DeclareTextCommandDefault{\textoneoldstyle}%
1095     {\tc@check@symbol2\textoneoldstyle}
1096 \DeclareTextCommandDefault{\texttwooldstyle}%
1097     {\tc@check@symbol2\texttwooldstyle}
1098 \DeclareTextCommandDefault{\textthreeoldstyle}%
1099     {\tc@check@symbol2\textthreeoldstyle}
1100 \DeclareTextCommandDefault{\textfouroldstyle}%
1101     {\tc@check@symbol2\textfouroldstyle}
1102 \DeclareTextCommandDefault{\textfiveoldstyle}%
1103     {\tc@check@symbol2\textfiveoldstyle}
1104 \DeclareTextCommandDefault{\textsixoldstyle}%
1105     {\tc@check@symbol2\textsixoldstyle}
1106 \DeclareTextCommandDefault{\textsevenoldstyle}%
1107     {\tc@check@symbol2\textsevenoldstyle}
1108 \DeclareTextCommandDefault{\texteightoldstyle}%
1109     {\tc@check@symbol2\texteightoldstyle}
1110 \DeclareTextCommandDefault{\textnineoldstyle}%
1111     {\tc@check@symbol2\textnineoldstyle}
1112 \DeclareTextCommandDefault{\textlangle}%
1113     {\tc@check@symbol2\textlangle}
1114 \DeclareTextCommandDefault{\textrightangle}%
1115     {\tc@check@symbol2\textrightangle}
1116 \DeclareTextCommandDefault{\textmho}%
1117     {\tc@check@symbol2\textmho}
1118 \DeclareTextCommandDefault{\textbigcircle}%
1119     {\tc@check@symbol2\textbigcircle}
1120 \DeclareTextCommandDefault{\textuparrow}%
1121     {\tc@check@symbol2\textuparrow}
1122 \DeclareTextCommandDefault{\textdownarrow}%
1123     {\tc@check@symbol2\textdownarrow}
1124 \DeclareTextCommandDefault{\textborn}%
1125     {\tc@check@symbol2\textborn}
1126 \DeclareTextCommandDefault{\textdivorced}%
1127     {\tc@check@symbol2\textdivorced}
1128 \DeclareTextCommandDefault{\textdied}%
1129     {\tc@check@symbol2\textdied}
1130 \DeclareTextCommandDefault{\textleaf}%
1131     {\tc@check@symbol2\textleaf}
1132 \DeclareTextCommandDefault{\textmarried}%
1133     {\tc@check@symbol2\textmarried}
1134 \DeclareTextCommandDefault{\textmusicalnote}%
1135     {\tc@check@symbol2\textmusicalnote}
1136 \DeclareTextCommandDefault{\textdblhyphenchar}%
1137     {\tc@check@symbol2\textdblhyphenchar}
1138 \DeclareTextCommandDefault{\textdollaroldstyle}%
1139     {\tc@check@symbol2\textdollaroldstyle}
1140 \DeclareTextCommandDefault{\textcentoldstyle}%

```

```

1141   {\tc@check@symbol2\textcentoldstyle}
1142 \DeclareTextCommandDefault{\textcolonmonetary}%
1143   {\tc@check@symbol2\textcolonmonetary}
1144 \DeclareTextCommandDefault{\textwon}%
1145   {\tc@check@symbol2\textwon}
1146 \DeclareTextCommandDefault{\textnaira}%
1147   {\tc@check@symbol2\textnaira}
1148 \DeclareTextCommandDefault{\textguarani}%
1149   {\tc@check@symbol2\textguarani}
1150 \DeclareTextCommandDefault{\textpeso}%
1151   {\tc@check@symbol2\textpeso}
1152 \DeclareTextCommandDefault{\textlira}%
1153   {\tc@check@symbol2\textlira}
1154 \DeclareTextCommandDefault{\textrecipe}%
1155   {\tc@check@symbol2\textrecipe}
1156 \DeclareTextCommandDefault{\textinterrobang}%
1157   {\tc@check@symbol2\textinterrobang}
1158 \DeclareTextCommandDefault{\textinterrobangdown}%
1159   {\tc@check@symbol2\textinterrobangdown}
1160 \DeclareTextCommandDefault{\textdong}%
1161   {\tc@check@symbol2\textdong}
1162 \DeclareTextCommandDefault{\textpertenthousand}%
1163   {\tc@check@symbol2\textpertenthousand}
1164 \DeclareTextCommandDefault{\textpilcrow}%
1165   {\tc@check@symbol2\textpilcrow}
1166 \DeclareTextCommandDefault{\textbaht}%
1167   {\tc@check@symbol2\textbaht}
1168 \DeclareTextCommandDefault{\textnumero}%
1169   {\tc@check@symbol2\textnumero}
1170 \DeclareTextCommandDefault{\textdiscount}%
1171   {\tc@check@symbol2\textdiscount}
1172 \DeclareTextCommandDefault{\textopenbullet}%
1173   {\tc@check@symbol2\textopenbullet}
1174 \DeclareTextCommandDefault{\textservicemark}%
1175   {\tc@check@symbol2\textservicemark}
1176 \DeclareTextCommandDefault{\textlquill}%
1177   {\tc@check@symbol2\textlquill}
1178 \DeclareTextCommandDefault{\textrquill}%
1179   {\tc@check@symbol2\textrquill}
1180 \DeclareTextCommandDefault{\textcopyleft}%
1181   {\tc@check@symbol2\textcopyleft}
1182 \DeclareTextCommandDefault{\textcircledP}%
1183   {\tc@check@symbol2\textcircledP}
1184 \DeclareTextCommandDefault{\textreferencemark}%
1185   {\tc@check@symbol2\textreferencemark}
1186 \DeclareTextCommandDefault{\textsurd}%
1187   {\tc@check@symbol2\textsurd}

```

The `\textcircled` and `\t` are handled specially, unless the current font has a subset id of 0 (i.e. full TS1) we pick the symbols up from the the math font encodings, i.e., the third argument to `\CheckEncodingSubset` uses `\UseTextAccent` to get them from there.

```

1188 \DeclareTextCommandDefault{\textcircled}
1189   {\CheckEncodingSubset\UseTextAccent{TS1}}%

```

```

1190   {\UseTextAccent{OMS}}1\textcircled}
1191 \DeclareTextCommandDefault{\t}
1192   {\CheckEncodingSubset\UseTextAccent{TS1}%
1193    {\UseTextAccent{OML}}1\t}

```

Finally input the encoding-specific definitions for TS1 thus making the top-level definitions optimised for this encoding (and not for the default encoding, see section 19.2).

```
1194 \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 (see section 19.1 above). So we better get rid of them:

```

1195 \UndeclareTextCommand{\textsterling}{OT1}
1196 \UndeclareTextCommand{\textdollar} {OT1}

```

Similar declarations should probably be made for other encodings like OT4 if they are in use.

```

1197 %\UndeclareTextCommand{\textsterling}{OT4}
1198 %\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.

```

1199 \UndeclareTextCommand{\textperthousand}{T1}
1200 %\UndeclareTextCommand{\textpertenthousand}{T1}

```

20.2.1 Supporting oldstyle digits

```

1201 \DeclareRobustCommand\oldstylenums[1]{%
1202   \begingroup
1203   \ifmmode
1204     \mathgroup\symletters #1%
1205   \else
1206     \CheckEncodingSubset\@use@text@encoding{TS1}%
1207     {\PackageWarning{textcomp}%
1208      {Oldstyle digits unavailable for
1209       family \f@family.\MessageBreak
1210        Lining digits used instead}}%
1211     \tw@{#1}%
1212   \fi
1213 \endgroup
1214 }

```

20.2.2 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:

```
1215 \iftc@forced \else
```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```
1216 \DeclareEncodingSubset{TS1}{cmr}      {0}
1217 \DeclareEncodingSubset{TS1}{cmss}     {0}
1218 \DeclareEncodingSubset{TS1}{cmtt}     {0}
1219 \DeclareEncodingSubset{TS1}{cmvtt}    {0}
1220 \DeclareEncodingSubset{TS1}{cmbr}     {0}
1221 \DeclareEncodingSubset{TS1}{cmtl}     {0}
1222 \DeclareEncodingSubset{TS1}{ccr}      {0}
```

PSNFSS fonts:

```
1223 \DeclareEncodingSubset{TS1}{ptm}      {4}
1224 \DeclareEncodingSubset{TS1}{pcr}      {4}
1225 \DeclareEncodingSubset{TS1}{phv}      {4}
1226 \DeclareEncodingSubset{TS1}{ppl}      {3}
1227 \DeclareEncodingSubset{TS1}{pag}      {4}
1228 \DeclareEncodingSubset{TS1}{pbk}      {4}
1229 \DeclareEncodingSubset{TS1}{pnc}      {4}
1230 \DeclareEncodingSubset{TS1}{pzc}      {4}
1231 \DeclareEncodingSubset{TS1}{bch}      {4}
1232 \DeclareEncodingSubset{TS1}{put}      {5}
```

Other CTAN fonts (probably not complete):

```
1233 \DeclareEncodingSubset{TS1}{uag}      {5}
1234 \DeclareEncodingSubset{TS1}{ugq}      {5}
1235 \DeclareEncodingSubset{TS1}{ul8}      {4}
1236 \DeclareEncodingSubset{TS1}{ul9}      {4} % (LuxiSans, one day)
1237 \DeclareEncodingSubset{TS1}{augie}    {5}
1238 \DeclareEncodingSubset{TS1}{dayrom}   {3}
1239 \DeclareEncodingSubset{TS1}{dayroms}  {3}
1240 \DeclareEncodingSubset{TS1}{pxr}      {0}
1241 \DeclareEncodingSubset{TS1}{pxss}     {0}
1242 \DeclareEncodingSubset{TS1}{pxtt}     {0}
1243 \DeclareEncodingSubset{TS1}{txr}      {0}
1244 \DeclareEncodingSubset{TS1}{txss}     {0}
1245 \DeclareEncodingSubset{TS1}{txtt}     {0}
```

Latin Modern and TeX Gyre:

```
1246 \DeclareEncodingSubset{TS1}{lmr}      {0}
1247 \DeclareEncodingSubset{TS1}{lmdh}     {0}
1248 \DeclareEncodingSubset{TS1}{lmss}     {0}
1249 \DeclareEncodingSubset{TS1}{lmssq}    {0}
1250 \DeclareEncodingSubset{TS1}{lmvtt}    {0}
1251 \DeclareEncodingSubset{TS1}{lmtt}     {0}

1252 \DeclareEncodingSubset{TS1}{qhv}      {0}
1253 \DeclareEncodingSubset{TS1}{qag}      {0}
1254 \DeclareEncodingSubset{TS1}{qbk}      {0}
```

```

1255 \DeclareEncodingSubset{TS1}{qcr}    {0}
1256 \DeclareEncodingSubset{TS1}{qcs}    {0}
1257 \DeclareEncodingSubset{TS1}{qpl}    {0}
1258 \DeclareEncodingSubset{TS1}{qtm}    {0}
1259 \DeclareEncodingSubset{TS1}{qzc}    {0}
1260 \DeclareEncodingSubset{TS1}{qhvc}    {0}

```

Fourier-GUTenberg:

```

1261 \DeclareEncodingSubset{TS1}{futs}    {4}
1262 \DeclareEncodingSubset{TS1}{futex}  {4}
1263 \DeclareEncodingSubset{TS1}{futj}    {4}

```

Y&Y's Lucida Bright

```

1264 \DeclareEncodingSubset{TS1}{hlh}    {3}
1265 \DeclareEncodingSubset{TS1}{hls}    {3}
1266 \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`.

```

1267 \DeclareEncodingSubset{TS1}{hlct}    {5}
1268 \DeclareEncodingSubset{TS1}{hlx}    {5}
1269 \DeclareEncodingSubset{TS1}{hlce}    {5}
1270 \DeclareEncodingSubset{TS1}{hlcn}    {5}
1271 \DeclareEncodingSubset{TS1}{hlcw}    {5}
1272 \DeclareEncodingSubset{TS1}{hlcf}    {5}

```

Other commercial families...

```

1273 \DeclareEncodingSubset{TS1}{pplx}    {3}
1274 \DeclareEncodingSubset{TS1}{pplj}    {3}
1275 \DeclareEncodingSubset{TS1}{ptmx}    {4}
1276 \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.

```

1277 \InputIfFileExists{textcomp.cfg}
1278 {\PackageInfo{textcomp}{Local configuration file used}}{}

1279 \fi
1280 </TS1sty>

```

File m

ltxcounts.dtx

21 Counters and Lengths

Commands for defining and using counters. This file defines:

<code>\newcounter</code>	To define a new counter.
<code>\setcounter</code>	To set the value of counters.
<code>\addtocounter</code>	Increase the counter #1 by the number #2.
<code>\stepcounter</code>	Increase a counter by one.
<code>\refstepcounter</code>	Increase a counter by one, also setting the value used by <code>\label</code> .
<code>\value</code>	For accessing the value of the counter as a T _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> : *, †, ‡, ...

1 (*2ekernel)

21.1 Environment Counter Macros

An environment foo has an associated counter defined by the following control sequences:

<code>\c@foo</code>	Contains the counter's numerical value. It is defined by <code>\newcount\foocounter</code> .
<code>\thefoo</code>	Macro that expands to the printed value of <code>\foocounter</code> . For example, if sections are numbered within chapters, and section headings look like Section II-3. The Nature of Counters then <code>\thesection</code> might be defined by: <code>\def\thesection</code> <code>{\@Roman{\c@chapter}-\@arabic{\c@section}}</code>
<code>\p@foo</code>	Macro that expands to a printed 'reference prefix' of counter foo. Any <code>\ref</code> to a value created by counter foo will produce the expansion of <code>\p@foo\thefoo</code> when the <code>\label</code> command is executed. See file <code>ltxref.dtx</code> for an extension of this mechanism.
<code>\cl@foo</code>	List of counters to be reset when foo stepped. Has format <code>\@elt{countera}\@elt{counterb}\@elt{counterc}</code> .

NOTE:

`\thefoo` and `\p@foo` *must* be defined in such a way that `\edef\bar{\thefoo}` or `\edef\bar{\p@foo}` defines `\bar` so that it will evaluate to the counter value at the time of the `\edef`, even after `\foocounter` and any other counters have been changed. This will happen if you use the standard commands `\@arabic`, `\@Roman`, etc.

The following commands are used to define and modify counters.

```

    \refstepcounter{<foo>}
    Same as \stepcounter, but it also defines \@currentreference so that a subsequent \label{<bar>} command causes \ref{<bar>} to generate the current value of counter <foo>.
    \@definecounter{<foo>}
    Initializes counter {<foo>} (with empty reset list), defines \p@foo and \thefoo to be null. Also adds <foo> to \cl@ckpt – the reset list of a dummy counter @ckpt used for taking checkpoints for the \include system.
    \@addtoreset{<foo>}{<bar>} : Adds counter <foo> to the list of counters \cl@bar to be reset when counter <bar> is stepped.

\setcounter \setcounter{<foo>}{<val>} : Globally sets \foocounter equal to <val>.
    2 \def\setcounter#1#2{%
    3   \@ifundefined{c@#1}%
    4     {\@nocounterr{#1}}%
    5     {\global\csname c@#1\endcsname#2\relax}}

\addtocounter \addtocounter{<foo>}{<val>} Globally increments \foocounter by <val>.
    6 \def\addtocounter#1#2{%
    7   \@ifundefined{c@#1}%
    8     {\@nocounterr{#1}}%
    9     {\global\advance\csname c@#1\endcsname #2\relax}}

\newcounter \newcounter{<newctr>}[<oldctr>] Defines <newctr> to be a counter, which is reset when counter <oldctr> is stepped. If <newctr> already defined produces ‘c@newctr already defined’ error.
    10 \def\newcounter#1{%
    11   \expandafter\ifdefinable \csname c@#1\endcsname
    12     {\@definecounter{#1}}%
    13     \@ifnextchar[{\@newctr{#1}}-]}

\value \value{<ctr>} produces the value of counter <ctr>, for use with a \setcounter or \addtocounter command.
    14 \def\value#1{\csname c@#1\endcsname}

\@newctr
    15 \def\@newctr#1[#2]{%
    16   \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}

\stepcounter \stepcounterfoo Globally increments counter \c@F00 and resets all subsidiary counters.
    17 \def\stepcounter#1{%
    18   \addtocounter{#1}\@ne
    19   \begingroup
    20     \let\@elt\@stpelt
    21     \csname cl@#1\endcsname
    22   \endgroup}

\@stpelt Rather than resetting the “within” counter to zero we set it to –1 and then run \stepcounter that moves it to 0 and also initiates resetting the next level down.
    23 \kernel
    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\endcsname\expandafter
43   {\expandafter\@arabic\csname c@#1\endcsname}}

\@addtoreset
44 \def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}}

Numbering commands for definitions of \theCOUNTER and \list arguments.
All commands can now be used in text and math mode.

\arabic Representation of counter as arabic numerals. Changed 29 Apr 86 to make it
print the obvious thing it COUNTER not positive.
45 \def\arabic#1{\expandafter\@arabic\csname c@#1\endcsname}

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

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

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

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

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

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

```



```

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

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

\@slowromancap Fully expandable macro to change a roman number to uppercase.
54 \def\@slowromancap#1{\ifx @#1% then terminate
55     \else
56     \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
57     c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi
58     \expandafter\@slowromancap
59     \fi
60 }

\@alph \@alph\F00counter Representation of \F00counter as a lower-case letter: 1 =
a, 2 = b, etc.
61 \def\@alph#1{%
62     \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
63     k\or l\or m\or n\or o\or p\or q\or r\or s\or t\or u\or v\or w\or x\or
64     y\or z\else\@ctrerr\fi}

\@Alph \@Alph\F00counter Representation of \F00counter as an upper-case letter: 1 =
A, 2 = B, etc.
65 \def\@Alph#1{%
66     \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
67     K\or L\or M\or N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or X\or
68     Y\or Z\else\@ctrerr\fi}

\@fnsymbol Typesetting old fashioned footnote symbols. This can be done both in text or
math mode now.

This macro is another example of an ever recurring problem in TEX: Deter-
mining if something is text-mode or math-mode. It is imperative for the decision
between text and math to be delayed until the actual typesetting is done as the
code in question may go through an \edef or \write where an \ifmode test
would be executed prematurely. Hence in the implementation below, \@fnsymbol
is not robust in itself but the parts doing the actual typesetting are.

In the case of \@fnsymbol we make use of the robust command \TextOrMath
which takes two arguments and typesets the first if in text-mode and the second if
in math-mode. Note that in order for this command to make the correct decision,
it must insert a \relax token if run under regular TEX, which ruins any kerning
between the preceding characters and whatever awaits typesetting. If you use
eTEX as engine for LATEX (as recommended) this unfortunate side effect is not
present.

69 </2ekernel>
70 <latexrelease>\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TextOrMath}%
71 <*2ekernel | latexrelease>
72 \def\@fnsymbol#1{%
73     \ifcase#1\or \TextOrMath\textasteriskcentered *\or

```

```

74 \TextOrMath \textdagger \dagger\or
75 \TextOrMath \textdaggerdbl \ddagger \or
76 \TextOrMath \textsection \mathsection\or
77 \TextOrMath \textparagraph \mathparagraph\or
78 \TextOrMath \textbardbl \|\or
79 \TextOrMath {\textasteriskcentered\textasteriskcentered}{**}\or
80 \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
81 \TextOrMath {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
82 \ctrerr \fi
83 }%
84 </2ekernel | latexrelease>
85 <latexrelease>\EndIncludeInRelease
86 <latexrelease>\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TextOrMath}%
87 <latexrelease>\def\@fnsymbol#1{\ensuremath{%
88 <latexrelease> \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
89 <latexrelease> \mathparagraph\or \|\or **\or \dagger\dagger
90 <latexrelease> \or \ddagger\ddagger \else\ctrerr\fi}}%
91 <latexrelease>\EndIncludeInRelease
92 <*2ekernel>

```

`\TextOrMath` When using regular \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 `\IeC` from `inputenc` but then it will have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of $e\TeX$ will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running $e\TeX$ but making sure not to permanently turn `\eTeXversion` into `\relax`.

```

93 </2ekernel>
94 <latexrelease>\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
95 <*2ekernel | latexrelease>
96 \begingroup\expandafter\expandafter\expandafter\endgroup
97 \expandafter\ifx\csname eTeXversion\endcsname\relax

```

In case of ordinary \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.

```

98 \DeclareRobustCommand\TextOrMath{%
99 \ifmode \expandafter\@secondoftwo
100 \else \expandafter\@firstoftwo \fi}
101 \protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
102 \else

```

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

```

103 \protected\expandafter\def\csname TextOrMath\space\endcsname{%
104 \ifmode \expandafter\@secondoftwo
105 \else \expandafter\@firstoftwo \fi}
106 \edef\TextOrMath#1#2{%
107 \expandafter\noexpand\csname TextOrMath\space\endcsname
108 {#1}{#2}}
109 \fi
110 </2ekernel | latexrelease>

```

```
111 <latexrelease>\EndIncludeInRelease
112 <latexrelease>\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
113 <latexrelease>\let\TextOrMath\@undefined
114 <latexrelease>\EndIncludeInRelease
115 <*2kernel>
116 </2kernel>
```

File n ltnlength.dtx

22 Lengths

```
\newlength Declare #1 to be a new length command.
\setlength Set the length command, #1, to the value #2.
\addtolength Increase the value of the length command, #1, by the value #2.
\settowidth Set the length, #1 to the width of a box containing #2.
\settoheight Set the length, #1 to the height of a box containing #2.
\settodepth Set the length, #1 to the depth of a box containing #2.

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

\newlength
3 \def\newlength#1{\@ifdefinable#1{\newskip#1}}

\setlength
4 (/2ekernel)
5 \langle latexrelease \rangle \IncludeInRelease{2015/01/01}%
6 \langle latexrelease \rangle \setlength}{Using \setlength with \dimen0}%
7 (*2ekernel | latexrelease)

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

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

\settoheight The obvious analogs of \settowidth.
\settodepth 17 \def\@settodim#1#2#3{\setbox\@tempboxa\hbox{##3}}#2#1\@tempboxa
\settowidth Clear the memory afterwards (which might be a lot).
\@settodim 18 \setbox\@tempboxa\box\voidb@x}
19 \def\settoheight{\@settodim\ht}
20 \def\settodepth {\@settodim\dp}
21 \def\settowidth {\@settodim\wd}

\@settopoint This macro takes the contents of the skip register that is supplied as its argument
and removes the fractional part to make it a whole number of points. This can be
used in class files to avoid values like 345.4666666pt when calculating a dimension.

22 \def\@settopoint#1{\divide#1\p@\multiply#1\p@}
23 (/2ekernel)
```

File o

ltfssbas.dtx

This file contains the main implementation of the ‘low level’ font selection commands. See other parts of the L^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.

The ‘2ekernel’ code ensures that a `\usepackage{autofss1}` is essentially ignored if a ‘full’ format is being used that has picture mode already in the format. Note the `autofss2` loading is currently disabled.

```
1 (2ekernel)\expandafter\let\csname ver@autofss1.sty\endcsname\fmtversion
```

23 Preliminary macros

We define a number of macros that will be used later.

`\@nomath` `\@nomath` is used by most macros that will have no effect in math mode. It issues a warning message.

```
2 (*2ekernel)
3 \def\@nomath#1{\relax\ifmmode
4   \@font@warning{Command \noexpand#1invalid in math mode}\fi}
```

`\no@alphabet@error` The macro `\no@alphabet@error` is called whenever the user requests a math *alphabet* that is not available in the current *version*. In math mode an error message is produced otherwise the command keeps silent. The argument is the name of the control sequence that identifies the math *alphabet*. The `\relax` at the beginning is necessary to prevent T_EX from scanning too far in certain situations.

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

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

```
\mathgroup
15 %\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@n}
16 \let\mathgroup\fam
17 %\let\newfam\new@mathgroup
18 \@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.

24 Macros for setting up the tables

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

```
19 \def\DeclareFontShape{\begingroup
```

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

```
20 \nfss@catcodes
```

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

`\DeclareFontShape`

```
21 \expandafter\endgroup
22 \DeclareFontShape@}
23 \def\DeclareFontShape@#1#2#3#4#5#6{%
24 \expandafter\ifx\csname #1+#2\endcsname\relax
25 \latex@error{Font family ‘#1+#2’ unknown}\@eha
26 \else
27 \expandafter
28 \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
29 \csname #5\endcsname}%
30 \def\reserved@a{#6}%
31 \global
32 \expandafter\let\csname#5\endcsname\expandafter\endcsname
33 \ifx\reserved@a\@empty
34 \@empty
35 \else
36 \reserved@a
37 \fi
38 \fi
39 }
```

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

```
40 \def\DeclareFixedFont#1#2#3#4#5#6{%
41 \begingroup
42 \math@fontsfalse
43 \every@math@size{}%
44 \fontsize{#6}\z@
45 \usefont{#2}{#3}{#4}{#5}%
46 \global\expandafter\let\expandafter#1\the\font
47 \endgroup
48 }
```

`\do@subst@correction`

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

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

```
56 \aftergroup\subst@correction
57 }
```

`\DeclareFontFamily`

```
58 \def\DeclareFontFamily#1#2#3{%  
If we want fast checking for the encoding scheme we can just check for \T@. . being  
defined.  
59 % \@tempwafalse  
60 % \def\reserved@b{#1}%  
61 % \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%  
62 %     \ifx\reserved@b\reserved@c \@tempwattrue\fi}%  
63 % \cdp@list  
64 % \if@tempswa  
65 % \ifundefined{T@#1}%  
66 %     {%  
67 %     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha  
68 %     }%  
69 %     {%
```

Now we have to define the macro `\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`.

```
70     \def\reserved@a{#3}%  
We compare \reserved@a with \@empty If these two are the same we \let the ‘extra’  
macro equal to \@empty which is not the same a doing a \let to \reserved@a  
— the latter would blow one extra memory location rather then reusing the one  
from \@empty.  
71     \global  
72     \expandafter\let\csname #1+#2\expandafter\endcsname  
73         \ifx \reserved@a\@empty  
74             \@empty  
75         \else \reserved@a  
76         \fi  
77     }%  
78 }
```

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

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

`\cdp@elt`

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

`\DeclareFontEncoding`

```
83 \def\DeclareFontEncoding{%
```

First we start with ignoring all blanks and newlines since every surplus space in the second or third argument will come out in a weird place in the document.

```
84     \begingroup  
85     \nfss@catcodes  
86     \expandafter\endgroup  
87     \DeclareFontEncoding@}  
88 \@onlypreamble\DeclareFontEncoding
```

```

89 \def\DeclareFontEncoding#1#2#3{%
90   \expandafter
91   \ifx\csname T@#1\endcsname\relax
92     \def\cdp@elt{\noexpand\cdp@elt}%
93     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
94                   {\default@family}{\default@series}%
95                   {\default@shape}}%

```

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

```

96     \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
97   \else
98     \@font@info{Redeclaring font encoding #1}%
99   \fi
100  \global\@namedef{T@#1}{#2}%
101  \global\@namedef{M@#1}{\default@M#3}%

```

Keep a record of the last encoding being declared:

```

102  \xdef\LastDeclaredEncoding{#1}%
103  }
104  \@onlypreamble\DeclareFontEncoding@

```

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

```

105 \def\LastDeclaredEncoding{}

```

`\DeclareFontSubstitution`

```

106 \def\DeclareFontSubstitution#1#2#3#4{%
107   \expandafter
108   \ifx\csname T@#1\endcsname\relax
109     \@latex@error{Encoding scheme ‘#1’ unknown}\@eha
110   \else
111     \begingroup

```

We loop through the `\cdp@list` and rebuild it anew in `\toks@` thereby replacing the defaults for the encoding in question with the new defaults. It is important to store the encoding to test against expanded in `\reserved@a` since it might just be `\LastDeclaredEncoding` that is passed as `#1`.

```

112     \edef\reserved@a{#1}%
113     \toks@{}%
114     \def\cdp@elt##1##2##3##4{%
115       \def\reserved@b{##1}%
116       \ifx\reserved@a\reserved@b

```

Here we use the new defaults but we use `##1` (i.e., the encoding name already stored previously) since we know that it is expanded.

```

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

```

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

```

119         \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
120       \fi}%
121     \cdp@list

```



```

122     \xdef\cdp@list{\the\toks@}%
123   \endgroup
124   \global
125   \@namedef{D@#1}{%
126     \def\default@family{#2}%
127     \def\default@series{#3}%
128     \def\default@shape{#4}%
129     }%
130   \fi
131 }
132 \@onlypreamble\DeclareFontSubstitution

```

\DeclareFontEncodingDefaults

```

133 \def\DeclareFontEncodingDefaults#1#2{%
134   \ifx\relax#1\else
135     \ifx\default@T\@empty\else
136       \@font@info{Overwriting encoding scheme text defaults}%
137     \fi
138     \gdef\default@T{#1}%
139   \fi
140   \ifx\relax#2\else
141     \ifx\default@M\@empty\else
142       \@font@info{Overwriting encoding scheme math defaults}%
143     \fi
144     \gdef\default@M{#2}%
145   \fi
146 }
147 \@onlypreamble\DeclareFontEncodingDefaults

```

\default@T

\default@M

```

148 \let\default@T\@empty
149 \let\default@M\@empty

```

\DeclarePreloadSizes

```

150 \def\DeclarePreloadSizes#1#2#3#4#5{%
151   \@ifundefined{T@#1}{%
152     {\@latex@error{Encoding scheme ‘#1’ unknown}\@eha}%
153     }%

```

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

```

154   \begingroup

```

We define a macro `\reserved@f`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).

```

155     \def\reserved@f##1,{%

```

The end of the list will be detected when there are no more elements, i.e. when `\reserved@f`'s argument is empty. The trick used here is explained in Appendix D of the *T_EXbook*: if the argument is empty the `\if` will select the first clause and `\let \reserved@f = \relax`. (We use the `>` character here since it cannot appear in font file names.)

```

156     \if>##1>%

```

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

```

157         \let\reserved@f\relax
158         \else

```

Otherwise, we define `\font@name` appropriately and call `\pickup@font` to do the work. Note that the requested `\curr@fontshape` combination must have been defined, or you will get an error. The definition of `\font@name` is carried out globally to be consistent with the rest of the code in this file.

```

159         \xdef\font@name{\csname#1/#2/#3/#4/##1\endcsname}%
160         \pickup@font

```

Now we forget the name of the font just loaded. More precisely, we set the corresponding control sequence to `\relax`. This means that later on, when the font is first used, the macro `\define@newfont` is called again to execute the ‘extra’ macro for this font.

```

161         \global\expandafter\let\font@name\relax
162         \fi

```

Finally we call `\reserved@f` again to process the next *size*. If `\reserved@f` was `\let` equal to `\relax` this will end the macro.

```

163         \reserved@f}%

```

We finish with reinserting the list of sizes after the `\reserved@f` macro and appending an empty element so that the end of the list is recognized properly.

```

164     \reserved@f#5,,%
165     \endgroup
166   }%
167 }
168 \@onlypreamble\DeclarePreloadSizes

```

`\ifmath@fonts` We need a switch to decide if we have to switch math fonts. For this purpose we provide `\ifmath@fonts` that can be set to true or false by the `\S@...` macros depending on if math fonts are provided for this size or not. The default is of course to switch all fonts.

```

169 \newif\ifmath@fonts \math@fontstrue

```

`\DeclareMathSizes` `\DeclareMathSizes` takes the text size, math text size, math script size, and math scriptscript size as arguments and defines the right `\S@...` macro.

```

\DeclareMathSizes*
170 \def\DeclareMathSizes{%
171   \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
172   {\@DeclareMathSizes{}}%
173 \@onlypreamble\DeclareMathSizes

```

`\@DeclareMathSizes` This modification by Michael J. Downes on `comp.text.tex` on 2002/10/17 allows the user to have settings such as `\DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}`.

```

174 </2ekernel>
175 <latexrelease>\IncludeInRelease{2015/01/01}{\@DeclareMathSizes}%
176 <latexrelease>           {Arbitrary units in \DeclareMathSizes}%
177 <*2ekernel|latexrelease>
178 \def\@DeclareMathSizes #1#2#3#4#5{%
179   \@defaultunits\dimen@ #2pt\relax\@nnil
180   \if $#3$%
181     \expandafter\let\csname S@\strip@pt\dimen@\endcsname\math@fontsfalse
182   \else

```

```

183 \defaultunits\dimen@ii #3pt\relax\@nnil
184 \defaultunits\@tempdima #4pt\relax\@nnil
185 \defaultunits\@tempdimb #5pt\relax\@nnil
186 \toks@{#1}%
187 \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
188   \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
189   \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
190   \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
191   \the\toks@
192 }%
193 \fi
194 }%
195 </2ekernel | latexrelease)
196 <latexrelease>\EndIncludeInRelease
197 <latexrelease>\IncludeInRelease{0000/00/00}{\@DeclareMathSizes}%
198 <latexrelease>           {Arbitrary units in \DeclareMathSizes}%
199 <latexrelease>\def\@DeclareMathSizes#1#2#3#4#5{%
200 <latexrelease>   \defaultunits\dimen@#2pt\relax\@nnil
201 <latexrelease>   \if$#3$%
202 <latexrelease>     \expandafter \let
203 <latexrelease>       \csname S@\strip@pt\dimen@\endcsname
204 <latexrelease>       \math@fontsfalse
205 <latexrelease>   \else
206 <latexrelease>     \expandafter \gdef
207 <latexrelease>       \csname S@\strip@pt\dimen@\endcsname
208 <latexrelease>         {\gdef\tf@size{#3}\gdef\sf@size{#4}%
209 <latexrelease>           \gdef\ssf@size{#5}%
210 <latexrelease>           #1%
211 <latexrelease>           }%
212 <latexrelease>   \fi}%
213 <latexrelease>\EndIncludeInRelease
214 <*2ekernel)
215 \onlypreamble\@DeclareMathSizes

```

25 Selecting a new font

25.1 Macros for the user

`\fontencoding` As we said in the introduction a font is described by four parameters. We first
`\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.

```

216 \DeclareRobustCommand\fontencoding[1]{%
217   \expandafter\ifx\csname T@#1\endcsname\relax
218     \@latexerror{Encoding scheme ‘#1’ unknown}\@eha
219   \else
220     \edef\f@encoding{#1}%
221     \ifx\cf@encoding\f@encoding

```

If the new encoding is the same as the old encoding we have nothing to do. However, in case we had a sequence of several encoding changes without a `\selectfont` in-between we can save processing by making sure that `\enc@update` is `\relax`.

```

222         \let\enc@update\relax
223     \else

```

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

```

224         \let\enc@update\@enc@update
225     \fi
226 \fi
227 }

```

`\@enc@update`

```

228 \def\@enc@update{%

```

When `\@enc@update` is executed `\f@encoding` holds the encoding name for the new encoding and `\cf@encoding` the name of the last active encoding.

We start by setting the init command for encoding dependent macros to `\@changed@cmd`.

```

229     \expandafter
230     \let
231     \csname\cf@encoding -cmd\endcsname
232     \@changed@cmd

```

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

```

233     \expandafter
234     \let
235     \csname\f@encoding-cmd\endcsname
236     \@current@cmd

```

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

```

237     \default@T
238     \csname T@\f@encoding\endcsname

```

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

```

239     \csname D@\f@encoding\endcsname
240     \let\enc@update\relax
241     \let\cf@encoding\f@encoding
242 }

```

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

```

243 \let\enc@update\relax

```

`\fontfamily`

```

\fontfamily 244 \DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
\fontseries 245 \DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
\fontseries 246 \DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}

```

`\fontshape` 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.

```

\fontshape 247 \def\usefont#1#2#3#4{\fontencoding{#1}\fontfamily{#2}%
\fontseries{#3}\fontshape{#4}\selectfont
248
249 \ignorespaces}

```

`\linespread` The command `\linespread` changes the current `\baselinestretch` by calling `\set@fontsize`. The values for `\f@size` and `\f@baselineskip` will be left unchanged.

```

250 \DeclareRobustCommand\linespread[1]
251   {\set@fontsize{#1}\f@size\f@baselineskip}

```

`\fontsize` We also define a macro that allows to specify a size. In this case, however, we also need the value of `\baselineskip`. As the first argument to `\set@fontsize` we pass the current value of `\baselinestretch`. This will either match the internal value (in which case nothing changes, or it will be an updated value due to a user change of that macro using `\renewcommand`. If we would pass the internal `\f@linespread` such a change would be effectively overwritten by a size change.

```

252 \DeclareRobustCommand\fontsize[2]
253   {\set@fontsize\baselinestretch{#1}{#2}}

```

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

```

254 \let\f@family\@empty
255 \let\f@series\@empty
256 \let\f@shape\@empty
257 \let\f@size\@empty
258 \let\f@baselineskip\@empty
259 \let\f@linespread\@empty

```

`\cf@encoding`

```

260 \let\f@encoding\@empty
261 \let\cf@encoding\@empty

```

`\@defaultunits` The function `\@defaultunits` when wrapped around a dimen or skip assignment supplies default units. Usage:

```

\@defaultunits\dimen@=#1pt\relax\@nnil

```

Note: the `\relax` is *important*. Other units can be substituted for the ‘pt’ if desired.

We use `\remove@to@nnil` as an auxiliary macros for `\@defaultunits`. It just has to gobble the supplied default unit ‘pt’ or whatever, if it wasn’t used in the assignment.

```

262 \def\@defaultunits{\afterassignment\remove@to@nnil}

```

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

```

\rem@pt
263 \begingroup
264   \catcode‘P=12
265   \catcode‘T=12
266   \lowercase{
267     \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}
268   \expandafter\endgroup\x
269 \def\strip@pt{\expandafter\rem@pt\the}

```

`\mathversion` `\mathversion` takes the math *version* name as argument, defines `\math@version` appropriately and switches to the font selected forcing a call to `\glb@settings` if the *version* is known to the system.

```

270 \DeclareRobustCommand\mathversion[1]
271   {\@nomath\mathversion

```

```

272     \expandafter\ifx\csname mv@#1\endcsname\relax
273     \latex@error{Math version '#1' is not defined}\@eha\else
274     \edef\math@version{#1}%

```

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

```

275     \gdef\glb@currsz{}%

```

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

```

276     \aftergroup\glb@settings
277     \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 278 \let\frozen@everymath\everymath
279 \let\frozen@everydisplay\everydisplay

\everymath Now we provide now user hooks that will be called in the frozen internals.
\everydisplay 280 \newtoks\everymath
281 \newtoks\everydisplay

\frozen@everymath Now we define the behaviour of the frozen hooks: first check the math setup then
call the user hook.
282 \frozen@everymath = {\check@mathfonts
283 \the\everymath}

\frozen@everydisplay Ditto for the display hook.
284 \frozen@everydisplay = {\check@mathfonts
285 \the\everydisplay}

\curr@math@size This holds locally the current math size.
286 \let\curr@math@size\@empty

```

25.2 Macros for loading fonts

- `\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.
- ```

287 \def\pickup@font{%
288 \expandafter \ifx \font@name \relax
289 \define@newfont
290 \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`
- ```

291 {\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.
- ```

292 \gdef\split@name#1/#2/#3/#4/#5\@nil{\def\f@encoding{#1}%
293 \def\f@family{#2}%
294 \def\f@series{#3}%
295 \def\f@shape{#4}%
296 \def\f@size{#5}}

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

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

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

298 \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.
- ```

299 \begingroup
300     \let\typeout\@font@info
301     \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`.
- ```

302 \expandafter\expandafter\expandafter
303 \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.

```

304 % \expandafter\ifx
305 % \csname\curr@fontshape\endcsname \relax
306 % \try@load@fontshape % try always
307 % \fi
308 \expandafter\ifx
309 \csname\curr@fontshape\endcsname \relax
310 \wrong@fontshape\else

```

To allow substitution we call the `curr@fontshape` macro which usually will expand to `\relax` but may hold code for substitution (see `\subst@fontshape` definition).

```

311 % \csname\curr@fontshape\endcsname
312 \extract@font\fi

```

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

```

313 \endgroup}
314 \def\try@load@fontshape{%
315 \expandafter
316 \ifx\csname \f@encoding+\f@family\endcsname\relax
317 \@font@info{Try loading font information for
318 \f@encoding+\f@family}%

```

We predefine this combination to be `\@empty` which means that next time we don't try again unnecessary in case we don't find a `.fd` file. If the file contains a `\DeclareFontFamily` command than this setting will be overwritten.

```

319 \global\expandafter\let
320 \csname\f@encoding+\f@family\endcsname\@empty

```

Set the catcodes used in the syntax, but do it only once (this will be restored at the end of the font loading group).

```

321 \nfss@catcodes
322 \let\nfss@catcodes\relax

```

For increased portability make the external filename monospace, but look for the (old style) mixed case filename if the first attempt fails.

On any monospace system this means that the file is looked for twice which takes up time and string space, but at least for this release Check for both names to give people time to re-install their private fd files with lowercase names.

```

323 \edef\reserved@a{%
324 \lowercase{%
325 \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}%
326 \reserved@a\relax
327 {\@input@\f@encoding\f@family.fd}}%
328 \fi}

```

`\nfss@catcodes` This macro should contain the standard `\catcode` assignments to all characters which are used in the commands found in an `.fd` file and which might have special `\catcodes` in the middle of a document. If necessary, this list can be extended in a package file using a suitable number of `\expandafter`, i.e.,

```

\expandafter\def\expandafter\nfss@catcodes
\expandafter{\nfss@catcodes <additional settings>}

```



Note, that this macro might get executed several times since it is also called by `\DeclareFontShape`, thus it probably should not be misused as a general purpose hook.

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

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

```
330 \makeatletter
331 \catcode'\ 9%
332 \catcode'\^^I9%
333 \catcode'\^^M9%
```

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

```
334 \catcode'\z@
335 \catcode'\{ \@ne
336 \catcode'\}\tw@
337 \catcode'\#6%
338 \catcode'\^7%
339 \catcode'\%14%
```

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

```
340 \@makeother\<%
341 \@makeother\>%
342 \@makeother*%
343 \@makeother\.%
344 \@makeother\-%
345 \@makeother\/%
346 \@makeother\[%
347 \@makeother\]%
348 \@makeother\'%
349 \@makeother\'%
350 \@makeother\"%
351 }
```

`\DeclareErrorFont` Declare the last resort shape! We assume that in this fontshape there is a 10pt font but it doesn't really matter. We only loose one macro name if the assumption is false. But at least the font should be there!

```
352 \def\DeclareErrorFont#1#2#3#4#5{%
353 \xdef\error@fontshape{%
354 \noexpand\expandafter\noexpand\split@name\noexpand\string
355 \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
356 \noexpand\@nil}%
```

Initialize all those internal variables which may or may not have values in the first seconds of NFSS' bootstrapping process. Later on such values will be updated when an encoding is selected, etc.

We definitely don't want to set `\f@encoding`; we can set all the others since if they are left "blank" any selection would grap "error default values" as well. However, this probably should go also.

```
357 % \gdef\f@encoding{#1}%
358 \gdef\default@family{#2}%
359 \gdef\default@series{#3}%
360 \gdef\default@shape{#4}%
361 \global\let\f@family\default@family
```

```

362 \global\let\f@series\default@series
363 \global\let\f@shape\default@shape
364 \gdef\f@size{#5}%
365 \gdef\f@baselineskip{#5pt}%
366 }
367 \onlypreamble\DeclareErrorFont

```

`\wrong@fontshape` Before we come to the macro `\extract@font` we have to take care of unknown `\curr@fontshape` combinations. The general strategy is to issue a warning and to try a default *shape*, then a default *series*, and finally a default *family*. If this last one also fails  $\TeX$  will go into an infinite loop. But if the defaults are set incorrectly one deserves nothing else!

```

368 (/2ekernel)
369 (latexrelease)\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
370 (latexrelease) {Font substitution in preamble}%
371 (*2ekernel | latexrelease)
372 \def\wrong@fontshape{%
373 \csname D@f@encoding\endcsname % install defaults if in math

```

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

```

374 \edef\reserved@a{\csname\curr@fontshape\endcsname}%
375 \ifx\last@fontshape\reserved@a
376 \errmessage{Corrupted NFSS tables}%
377 \error@fontshape
378 \else

```

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

```

379 \let\f@shape\default@shape

```

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

```

380 \expandafter\ifx\csname\curr@fontshape\endcsname\relax
381 \let\f@series\default@series

```

If this is still undefined, try the default *family*. Otherwise give up. We never try to change the encoding scheme!

```

382 \expandafter
383 \ifx\csname\curr@fontshape\endcsname\relax
384 \let\f@family\default@family

```

If we change the font family and we are in the preamble then the corresponding `.fd` file may not been loaded yet. Therefore we try this now. Otherwise equating the requested font shape with the finally selected fontshape below will fail and can result in “NFSS tables corrupted”. After begin document that will not happen as all `.fd` files involved in substitution are loaded at `\begin{document}`.

```

385 \begingroup
386 \try@load@fontshape
387 \endgroup
388 \fi \fi
389 \fi

```

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

The `\expandafter\string` here stops  $\TeX$  adding the space that it usually puts after command names in messages. The similar construction with `\undefined` just produces ‘undefined’, but saves a few tokens.

`\@wrong@font@char` is locally redefined in `\UseTextSymbol` from its normal (empty) definition, to report the symbol generating the font switch.

```
390 \font@warning{Font shape ‘\expandafter\string\reserved@a’
391 \expandafter@gobble\string@undefined\MessageBreak
392 using ‘\curr@fontshape’ instead@\wrong@font@char}%
393 \global\let\last@fontshape\reserved@a
```

We change `\@defaultsubs` to produce a warning at the end of the document.

The macro `\@defaultsubs` is initially `\relax` but gets changed here if some default font substitution happens. It is then executed in `\enddocument`.

```
394 \gdef\@defaultsubs{%
395 \font@warning{Some font shapes were not available, defaults
396 substituted.\gobbletwo}}%
```

If we substitute a `\curr@fontshape` combination by the default one we don't want the warning to be printed out whenever this (unknown) combination is used. Therefore we globally `\let` the macro corresponding to the wanted combination equal to its substitution. This requires the use of four `\expandafter`'s since `\csname... \endcsname` has to be expanded before `\reserved@a` (i.e. the requested combination), and this must happen before the `\let` is executed.

```
397 \global\expandafter\expandafter\expandafter\let
398 \expandafter\reserved@a
399 \csname\curr@fontshape\endcsname
```

Now we can redefine `\font@name` accordingly. This *must* be done globally since it might occur in the group opened by `\define@newfont`. If we would this definition were local the closing `\endgroup` there would restore the old meaning of `\font@name` and then switch to the wrong font at the end of `\selectfont` although the correct font was loaded.

```
400 \xdef\font@name{%
401 \csname\curr@fontshape/\f@size\endcsname}%
```

The last thing this macro does is to call `\pickup@font` again to load the font if it is not defined yet. At this point this code will loop endlessly if the defaults are not well defined.

```
402 \pickup@font}
403 </2ekernel | latexrelease)
404 (latexrelease)\EndIncludeInRelease
405 (latexrelease)\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
406 (latexrelease) {Font substitution in preamble}%
407 (latexrelease)\def\wrong@fontshape{%
408 (latexrelease) \csname D@\f@encoding\endcsname
409 (latexrelease) \edef\reserved@a{\csname\curr@fontshape\endcsname}%
410 (latexrelease) \ifx\last@fontshape\reserved@a
411 (latexrelease) \errmessage{Corrupted NFSS tables}%
412 (latexrelease) \error@fontshape
413 (latexrelease) \else
414 (latexrelease) \let\f@shape\default@shape
415 (latexrelease) \expandafter\ifx\csname\curr@fontshape\endcsname\relax
416 (latexrelease) \let\f@series\default@series
417 (latexrelease) \expandafter
418 (latexrelease) \ifx\csname\curr@fontshape\endcsname\relax
419 (latexrelease) \let\f@family\default@family
420 (latexrelease) \fi \fi
```

```

421 (latexrelease) \fi
422 (latexrelease) \@font@warning{Font shape
423 (latexrelease) '\expandafter\string\reserved@a'
424 (latexrelease) \expandafter\@gobble\string\@undefined
425 (latexrelease) \MessageBreak
426 (latexrelease) using '\curr@fontshape' instead\@wrong@font@char}%
427 (latexrelease) \global\let\last@fontshape\reserved@a
428 (latexrelease) \gdef\@defaultsubs{%
429 (latexrelease) \@font@warning{Some font shapes were not available,
430 (latexrelease) defaults substituted.\@gobbletwo}}%
431 (latexrelease) \global\expandafter\expandafter\expandafter\let
432 (latexrelease) \expandafter\reserved@a
433 (latexrelease) \csname\curr@fontshape\endcsname
434 (latexrelease) \xdef\font@name{%
435 (latexrelease) \csname\curr@fontshape/\f@size\endcsname}%
436 (latexrelease) \pickup@font}
437 (latexrelease)\EndIncludeInRelease
438 (*2kernel)

```

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

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

`\@defaultsubs` See above.

```
\@defaultsubs 440 \let\@defaultsubs\relax
```

`\strip@prefix` In `\extract@font` we will need a way to recover the replacement text of a macro. This is done by the primitive `\meaning` together with the macro `\strip@prefix` (for the details see appendix D of the *TeXbook*, p. 382).

```
441 \def\strip@prefix#1>{}
```

## 26 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

```
442 \let\install@mathalphabet\gdef
```

`\math@fonts`

```
443 \let\math@fonts\@empty
```

`\select@group` `\select@group` has four arguments: the new *math alphabet identifier* (a control sequence), the *math group number*, the extra macro for math mode and the `\curr@fontshape` definition macro name. We first check if we are in math mode.

```
444 %\def\select@group#1#2#3{\relax\ifmmode
```

We do these things locally using `\begingroup` instead of `\bgroup` to avoid the appearance of an empty Ord atom on the math list.

```
445 % \begingroup
```

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

```
446 % \escapechar\m@ne
```

```
447 % \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
```

We globally select the math fonts...

```
448 % \globaldefs\@ne \math@fonts
```

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

```
449 % \endgroup
```

As long as no *size* or *version* change occurs the  $\langle\math\text{alphabet identifier}\rangle$  should simply switch to the installed *math group* instead of calling `\select@group` unnecessarily. So we globally redefine the first argument (the new  $\langle\math\text{alphabet identifier}\rangle$ ) 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  $\langle\math\text{alphabet identifier}\rangle$  and then calling the new definition to switch to the wanted  $\langle\math\text{group}\rangle$ . Now we define the  $\langle\math\text{alphabet identifier}\rangle$  as a call to the `\use@mathgroup` command.

```
450 % \xdef#1{\noexpand\use@mathgroup\noexpand#2%
```

```
451 % {\number\csname c@mv@\math@version\endcsname}}%
```

But this is not sufficient, as we learned the hard way. The problem here is that the loading of the fonts that comprise the alphabet identifier `#1`, as well as the necessary math font assignments is deferred until it is used. This is OK so far, but if the fonts are switched within the current formula (which may happen if a sub-formula is a box that contains a math version switch) the font assignments for `#1` are not restored unless `#1` is used again. This is disastrous since TeX sees the wrong fonts at the end of the math formula, when it converts the math list into a horizontal list.

This is taken into account as follows: When a math alphabet identifier is used for the first time in a certain version it modifies the corresponding macro `\mv@<version>` so that it calls `\getanddefine@fonts` directly in future as well. We use the macro `\extract@alph@from@version` to do this. It takes the math alphabet identifier `#1` and the math version macro as arguments.

```
452 % \expandafter\extract@alph@from@version
```

```
453 % \csname mv@\math@version\expandafter\endcsname
```

```
454 % \expandafter{\number\csname c@mv@\math@version\endcsname}%
```

```
455 % #1%
```

```
456 % \stepcounter{mv@\math@version}%
```

Finally, it is not possible to simply call the new definition since we have an argument (the third argument of `\use@mathgroup` or more exactly the argument of `\math@egroup` if the `margid` option is in force) which would swallow our closing `\fi`. So we use the `\expandafter` technique to remove the `\fi` before the `\use@mathgroup` is expanded.

```
457 %\expandafter #1\fi}
```

```
\extract@alph@from@version
```

We proceed to the definition of the macro `\extract@alph@from@version`. As stated above, it takes a math alphabet identifier and a math version macro (e.g. `\mv@normal`) as its arguments.

```
458 \def\extract@alph@from@version#1#2#3{%
```

To extract and replace the definition of math alphabet identifier `#3` in macro `#1` we have to recall how this definition looks like: Somewhere in the replacement

text of #1 there is the sequence

```
\install@mathalphabet<math alphabet identifier> #3{%
 <Definitions for >#3}
```

Hence, the first thing we do is to extract the tokens preceding this definitions, the definition itself, and the tokens following it. To this end we define one auxiliary macro `\reserved@a`.

```
459 \def\reserved@a##1\install@mathalphabet#3##2##3\@nil{%
```

When `\reserved@a` is expanded, it will have the tokens preceding the definition in question in its first argument (`##1`), the following tokens in its third argument (`##3`), and the replacement text for the math alphabet identifier `#3` in its second argument. (`##2`). This is then recorded for later use in a temporary macro `\reserved@b`.

```
460 \def\reserved@b{##2}%
```

Additionally, we define a macro `\reserved@c` to reconstruct the definitions for the math version in question from the tokens that will remain unchanged (`##1` and `##3`) and the yet to build new definitions for the math alphabet identifier `#3`.

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

Then we execute our auxiliary macro.

```
462 \expandafter\reserved@a#1\@nil
```

OK, so now we have to build the new definition for `#3`. To do so, we first extract the interesting parts out of the old one. The old definition looks like:

```
\select@group<math alphabet identifier>
 <math group number><math extra part>
<curr@fontshape definition>
```

So we define a new temporary macro `\reserved@a` that extracts these parts.

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

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

```
464 \reserved@c{%
465 \getanddefine@fonts{#2}##2%
466 \install@mathalphabet#3{%
467 \relax\ifmmode \else \non@alpherr#3\fi
468 \use@mathgroup##1{#2}}}%
```

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

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

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

```
471 \expandafter\reserved@a\reserved@b\@nil
472 }
```

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

```
473 \let\math@bgroup\bgroup
474 \def\math@egroup#1{#1\egroup}
```

`\calculate@math@sizes` Here is the default definition for `\calculate@math@sizes` a more elaborate interface is under testing in `mthscale.sty`.

```
475 \gdef\calculate@math@sizes{%
476 \@font@info{Calculating\space math\space sizes\space for\space
477 size\space <\f@size>}%
478 \dimen@f@size \p@
479 \@tempdimb \defaultscritratio \dimen@
480 \dimen@ \defaultscritscritratio \dimen@
481 \expandafter\xdef\csname S@\f@size\endcsname{%
482 \gdef\noexpand\tf@size{\f@size}%
483 \gdef\noexpand\sfs@size{\strip@pt\@tempdimb}%
484 \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
485 \noexpand\math@fontstrue}}
```

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

```
486 \def\defaultscritratio{.7}
487 \def\defaultscritscritratio{.5}
```

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

```
488 \ifx\noaccents@\undefined
489 \let\noaccents@\@empty
490 \fi
```

`\showhyphens` The `\showhyphens` command must be redefined since the version in `plain.tex` uses `\tenrm`. We have also made some further adjustments for its use in L<sup>A</sup>T<sub>E</sub>X.

```
491 \gdef\showhyphens#1{%
492 \setbox0\vbox{%
493 \color@begingroup
494 \everypar{}%
495 \parfillskip\z@skip\hsize\maxdimen
496 \normalfont
497 \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@ #1%
498 \color@endgroup}}
```

`\addto@hook` We need a macro to add tokens to a hook.

```
499 \long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}
```

`\@vpt`

```
500 \def\@vpt{5}
```

`\@vipt`

```
501 \def\@vipt{6}
```

`\@viipt`

```
502 \def\@viipt{7}
```

`\@viiipt`

```
503 \def\@viiipt{8}
```

`\@ixpt`

```
504 \def\@ixpt{9}
```

```
\@xpt
505 \def\@xpt{10}

\@xipt
506 \def\@xipt{10.95}

\@xiipt
507 \def\@xiipt{12}

\@xivpt
508 \def\@xivpt{14.4}

\@xvipt
509 \def\@xvipt{17.28}

\@xxpt
510 \def\@xxpt{20.74}

\@xxvpt
511 \def\@xxvpt{24.88}
512 </2kernel>
```



# File p

## lfsstrc.dtx

### 27 Introduction

This package contains the code for tracing font loading and font changes. It basically overlays some of the low-level functions of NFSS with additional code used for tracing.

The package accepts the following options:

**errorshow** Write all information about font changes etc. only to the transcript file unless an error happens. This means that information about font substitution will not be shown on the terminal.

**warningshow** Show all NFSS warnings on the terminal. This setting corresponds to the default behaviour of NFSS if the `tracefnt` package is *not* loaded!

**infoshow** Show all NFSS warning and all NFSS info messages (that are normally only written to the transcript file) also on the terminal. This is the default if the `tracefnt` package is loaded.

**debugshow** In addition to `infoshow` show also changing of math fonts as far as possible (this option can produce a large amount of output).

**loading** Show the name of external fonts when they are loaded. This option shows only “newly” loaded fonts not those already preloaded in the format or the class file before the `tracefnt` package became active.

**pausing** Turn all font warnings into errors so that  $\text{\LaTeX}$  will stop.

### 28 A driver for this document

The next bit of code contains the documentation driver file for  $\text{\TeX}$ , i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

When this file is processed directly by  $\text{\LaTeX}$  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>
```

## 29 The Implementation

**Warning:** Read the macro documentation with a grain of salt. It is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

If we are making a package file it is a good idea to test whether we are running under 2e. This code is actually placed at the very beginning of this file for easier maintenance, thus commented out here.

```
11 (*package)
12 %\NeedsTeXFormat{LaTeX2e}
13 %\ProvidesPackage{tracefnt}[??/??/?? v?.??]
14 %
15 \input{Standard LaTeX package (font tracing)}
```

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

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

## 30 Handling Options

`\tracingfonts` Here is the definition of the integer register for the font trace. As a default in a package file we use 1 to give error messages if fonts are substituted. If this code is used for debugging or tracing reasons in the format file (i.e. in `fam.dtx`) we use 0 as the default. But if no font trace is used we build a definition that will produce a warning message.

```
17 (*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 \end{kernel}
```

The `\count@` in the line above will remove the number after `\tracingfonts`. Note that this definition will be overwritten by the next line if one of these modules are included.

```
24 (*package, trace, debug)
25 \newcount\tracingfonts
26 \tracingfonts=0
27 \end{package, trace, debug}
```

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

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

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

```

32 {LaTeX Font Info: \space\space\space#1}}%
33 \def\@font@warning#1{%
34 \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
35 {LaTeX Font Warning: #1}}%
36 }
37 \DeclareOption{warningshow}{%
38 \def\@font@info#1{%
39 \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
40 {LaTeX Font Info: \space\space\space#1}}%
41 \def\@font@warning#1{%
42 \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
43 {LaTeX Font Warning: #1}}%
44 }
45 \DeclareOption{infoshow}{%
46 \def\@font@info#1{%
47 \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
48 {LaTeX Font Info: \space\space\space#1}}%
49 \def\@font@warning#1{%
50 \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
51 {LaTeX Font Warning: #1}}%
52 }
53 \DeclareOption{loading}{%
54 \tracingfonts\tw@
55 }
56 \DeclareOption{debugshow}{%
57 \ExecuteOptions{infoshow}%
58 \tracingfonts\thr@@
59 }
60 \DeclareOption{pausing}{%
61 \def\@font@warning#1{%
62 \GenericError
63 {(Font)\@spaces\@spaces\@spaces\space\space}%
64 {LaTeX Font Warning: #1}%
65 {See the LaTeX Companion for details.}%
66 {I'll stop for every LaTeX Font Warning because
67 you requested\MessageBreak the 'pausing' option
68 to the tracefmt package.}}%
69 }

```

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

```

70 \ExecuteOptions{infoshow}
71 \ProcessOptions
72 </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)
```

## 31 Macros common to fam.tex and tracefnt.sty

In the first versions of `tracefnt.dtx` some macros of `fam.dtx`<sup>5</sup> 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`.

### 31.1 General font loading

`\extract@font` This macro organizes the font loading. It first calls `\get@external@font` which will return in `\external@font` the name of the external font file (the `.tfm`) as it was determined by the NFSS tables.

```
81 (*2ekernel | package)
82 \def\extract@font{%
83 \get@external@font
```

Then the external font is loaded and assigned to the font identifier stored inside `\font@name` (for this reason we need `\expandafter`).

```
84 \global\expandafter\font\font@name\external@font\relax
```

When tracing we typeout the internal and external font name.

```
85 (*trace)
86 \ifnum \tracingfonts >\@ne
87 \@font@info{External font '\external@font'
88 loaded as\MessageBreak \font@name}\fi
89 </trace)
```

Finally we call the corresponding “loading action” macros to finish things. First the font is locally selected to allow the use of `\font` inside the loading action macros.

```
90 \font@name \relax
```

The next two lines execute the “loading actions” for the family and then for the individual font shape.

```
91 \csname \f@encoding+\f@family\endcsname
92 \csname\curr@fontshape\endcsname
93 \relax
94 }
95 </2ekernel | package)
```

The `\relax` at the end needs to be explained. This is inserted to prevent `TEX` from scanning too far when it is executing the replacement text of the loading code macros.

`\get@external@font` This function tries to find an external font name. It will place the name into the macro `\external@font`. If no font is found it will return the one that was defined via `\DeclareErrorFont`.

```
96 (*2ekernel)
97 \def\get@external@font{%
```

<sup>5</sup>This file is currently not distributed in documented form. Its code is part of `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<sup>A</sup>T<sub>E</sub>X's `\twfbbf`, 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>
```

## 31.2 Math fonts setup

### 31.2.1 Outline of algorithm for math font sizes

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



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

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

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

This is handled in the following way:

1. At any point in the document the global variable `\gbl@currsize` 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@currsize`.
  - (a) If we are recursively inside another formula (`\if@inmath`) we ensure that `\gbl@settings` is executed again in the outer formula, so that the old setup is automatically restored.
  - (b) Otherwise, we set the switch `@inmath` locally to `true` so that all nested formulae will be able to detect that they are nested in some outer formula.

The above algorithm has the following features:

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

### 31.2.2 Code for math font size setting

`\check@mathfonts` In the `\check@mathfonts` macros we implement the steps 2 to 4 except that instead of a switch the macro `\init@restore@glb@settings` is used.

```

204 (*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 }

```

### 31.2.3 Other code for math

`\use@mathgroup` The `\use@mathgroup` macro should be used in user macros to select a math group. Depending on whether or not the `margid` option is in force it has two or three arguments. For this reason it should be called as the last macro.

First we test if we are inside math mode since we don't want to apply a useless definition.

```

253 \def\use@mathgroup#1#2{\relax\ifmmode
254 (*trace)
255 \ifnum \tracingfonts>\tw@
256 \count@#2\relax
257 \@font@info{Using \noexpand\mathgroup
258 (\the\count@) #2}\fi
259 }/trace)

```

If so we first call the '=' macro (i.e. argument three) to set up special things for the selected math group. Then we call `\mathgroup` to select the group given by argument two and finally we place #1 (i.e. the argument of the *math alphabet identifier*) at the end. This part of the code is surrounded by two commands which behave like `\begingroup` and `\endgroup` if we want *math alphabet identifier*s but will expand into `\empty` if we want simply switches to a new math group. Since argument number 2 may be a digit instead of a control sequence we add a `\relax`. Otherwise something like `\mit{1}` would switch to math group 11 (and back) instead of printing an oldstyle 1.

```

260 \math@bgroup
261 \expandafter\ifx\csname M@f@encoding\endcsname#1\else
262 #1\fi
263 \mathgroup#2\relax

```

Before we reinsert the swallowed token (arg. three) into the input stream, in the case that the *math alphabet identifier* isn't called in math mode, we remove the `\fi` with the `\expandafter` trick. This is necessary if the token is actually 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  $\TeX$  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.<sup>6</sup> Again, `font@name` is defined globally, for the reasons explained in the description of `\wrong@fontshape`.

```

284 \xdef\font@name{\csname \string#2/\tf@size\endcsname}%

```

Then we call `\pickup@font` to load it if necessary. We remember the internal name as `\textfont@name`.

```

285 \pickup@font \let\textfont@name\font@name

```

Same game for `\scriptfont` and `\scriptscriptfont`:

```

286 \xdef\font@name{\csname \string#2/\sf@size\endcsname}%
287 \pickup@font \let\scriptfont@name\font@name
288 \xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
289 \pickup@font

```

Then we append the new `\textfont...` assignments to the `\math@fonts`.

```

290 \edef\math@fonts{\math@fonts
291 \textfont#1\textfont@name
292 \scriptfont#1\scriptfont@name
293 \scriptscriptfont#1\font@name}%

```

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

```

294 (+debug) \poptracing
295 }
296 </2ekernel | package>

```

---

<sup>6</sup>One might ask why this expansion does not generate a macro name that starts with an additional `\` character. The solution is that `\escapechar` is set to `-1` before `\getanddefine@fonts` is called.

## 32 Scaled font extraction

`\ifnot@nil` We begin with a simple auxiliary macro. It checks whether its argument is the token `\@nil`. If so, it expands to `\@gobble` which discards the following argument, otherwise it expands to `\@firstofone` which reproduces its argument.

```
297 (*2kernel)
298 \def\ifnot@nil#1{\def\reserved@a{#1}%
299 \ifx\reserved@a\@nnil \expandafter\@gobble
300 \else \expandafter\@firstofone\fi}
```

`\remove@to@nnil` Three other auxiliary macros will be needed in the following: `\remove@to@nnil` gobbles up everything up to, and including, the next `\@nnil` token, and `\remove@angles` and `\remove@star` do the same for the character `>` and `*`, respectively, instead of `\@nnil`.

```
301 \def\remove@to@nnil#1\@nnil{}
302 \def\remove@angles#1>{\set@simple@size@args}
303 \def\remove@star#1*{#1}
```

`\extract@sizefn` This macro takes a size specification and parses it into size function and the optional and mandatory arguments.

```
304 \def\extract@sizefn#1*#2\@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 #3 will be 17>cmss17. The expansion is therefore

```
\set@simple@size@args cmss12<17*>cmss17\@nnil
\execute@size@function{}
```

This means: the default (empty) size function will be executed, with its optional argument argument set to empty and its mandatory argument set to cmss12 by \set@simple@size@args. As we discussed earlier, the effect of the default size function is to load the given external font (cmss12) at the specified size (12)—which is exactly what was intended.

But this is only part of the whole story. It may be that the size requested does not occur in the token list \font@info. And the simple definition of \extract@fontinfo we gave above does not allow to specify give more than one size specification in front of the external font name.

Let's address these two problems separately. The first one is solved with the following trick: We define \extract@fontinfo as follows:

```
\def\extract@fontinfo#1<12*#2>#3<#4\@nnil{%
 \ifnot@nil{#3}%
 {\set@simple@size@args#3<#4\@nnil
 \execute@size@function{#2}%
 }}%
```

How does this work? We call \extract@fontinfo via

```
\expandafter\extract@fontinfo\font@info<12*>\@nil<\@nnil
```

i.e. by appending <12\*>\@nil<\@nnil. If the size ('12' in this case) appears in \font@info everything works as explained above, the only difference being that argument #4 of \extract@fontinfo additionally gets the tokens <12\*>\@nil<\@nnil. However, if the size is not found everything up to the final <12\*> is in argument #1, #3 gets \@nil, and #2 and #4 are empty. The macro \ifnot@nil will discard the calls to \set@simple@size@args and execute@size@function, and hence \font@info will continue to be equal to \@empty. This means that no simple size specification matching the requested size could be found.

The second problem (more than one simple size specification for one external font name) will be addressed in \set@simple@size@args below.

The macros are hidden inside other control sequences so that we have to build \extract@fontinfo in several steps.

So here's the actual definition of \extract@font in \try@simple@size.

```
310 % % this could be replaced by \try@size@range making the subst slower!
```

```
311 \def\try@simple@size{%
```

\reserved@a is made an abbreviation for the head of the definition of the macro \extract@fontinfo.

```
312 \def\reserved@a{\def\extract@fontinfo###1}%
```

Now we can define \extract@fontinfo. Here we handle a small but convenient variation: in case of the default (empty) size function it is allowed to omit the \* character.

```
313 \expandafter\reserved@a\expandafter<\f@size>##2<##3\@nnil{%
```

```
314 \ifnot@nil{##2}%
```

```

315 {\set@simple@size@args##2<##3\@nnil
316 \execute@size@function\sizefn@info
317 }}%

```

Now we call `\extract@fontinfo`. Note the `<\@nil` tokens at the end.

```

318 \expandafter\expandafter
319 \expandafter\extract@fontinfo\expandafter\font@info
320 \expandafter<\f@size>\@nil<\@nnil
321 }

```

`\set@simple@size@args` As promised above, the macro `\set@simple@size@args` will handle the case of several size specifications in a row. If another size specification follows, the very first token of its argument list is the character `<`. By starting the definition as follows,

```

322 \def\set@simple@size@args#1<{%

```

parameter `#1` is empty in this case, and contains the size function's arguments otherwise. We distinguish these two cases (Note that the character `<` cannot appear in `#1`) by calling `\remove@angles` for empty `#1` and `\extract@sizefn` otherwise. In the latter case we have to take care of the remaining character tokens and discard them. This is done by `\remove@to@nnil`. Note also the use of Kabelschacht's method.

```

323 \if<#1<%
324 \expandafter\remove@angles
325 \else
326 \extract@sizefn#1*\@nil
327 \expandafter\remove@to@nnil
328 \fi}

```

Now, we are through with the case of a simple size, except for calling the size function. This will be handled later, as it is the same mechanism for all types of size specification. We will now proceed to macros for extraction of size range specification.

`\extract@rangefontinfo` `\extract@rangefontinfo` goes through a font shape definition in the input until it recognizes the tokens `<\@nil->`. It looks for font ranges with font size functions. It's operation is rather simple: it discards everything up to the next size specification and passes this on to `\is@range` for inspection. The specification (parameter `#2` is inserted again, in case it is needed later.

```

329 \def\extract@rangefontinfo#1<#2>{%
330 \is@range#2->\@nil#2>}

```

`\is@range` `\is@range` is again a sort of dispatcher macro: if the size specification it is looking at is not a range specification it discards it and calls `\extract@rangefontinfo` to continue the search. Otherwise it calls `\check@range` to check the requested size against the specified range.

From the way `\is@range` is called inside `\extract@rangefontinfo` we see that `#2` is the character `>` if the size specification found is a simple one (as it does not contain a `-` character. This is checked easily enough and `\extract@rangefontinfo` called again. Note that the extra tokens inserted after the `\@nil` in the call to `\is@range` appear at the beginning of the first argument to `\extract@rangefontinfo` and are hence ignored.

```

331 \def\is@range#1-#2\@nil{%
332 \if>#2\expandafter\check@single\else
333 \expandafter\check@range\fi}

```

`\check@range` `\check@range` takes lower bound as parameter #1, upper bound as #2, size function as #3 and the size function's arguments as #4. If #3 is the special token `\@nil` `\font@info` is exhausted and we can stop searching.

```

334 \def\check@range#1-#2>#3<#4\@nnil{%
335 \ifnot@nil{#3}{%

```

If #3 wasn't `\@nil` we have a range. We start by assuming that we have to recurse. Note that we have to reinsert an `<` as it was already removed by scanning.

```

336 \def\reserved@f{\extract@rangefontinfo<#4\@nnil}%

```

We have to make sure that both boundaries are present, if not we have to set them. Here we check the upper bound. If `\upper@bound` is zero after the assignment we set it to `\maxdimen` (upper open range). We need to use a *dimen* register for the scan since we may have a decimal number as the boundary.

```

337 \upper@bound0#2\p@
338 \ifdim\upper@bound=\z@ \upper@bound\maxdimen\fi

```

Now we check the upper boundary against `\f@size`. If it is larger or equal than `\f@size` this range is no good and we have to recurse.

```

339 \ifdim \f@size \p@<\upper@bound

```

Otherwise we have to check the lower bound. This time it is not necessary to scan the boundary value into a register because if it is empty we get zero as desired. We could even omit the 0 which would result in `1pt` as default lower boundary. If `\f@size` is smaller than the boundary we have to recurse.

```

340 \lower@bound0#1\p@
341 \ifdim \f@size \p@<\lower@bound
342 \else

```

If both tests are passed we can try executing the size function.

```

343 \set@simple@size@args#3<#4\@nnil
344 \execute@size@function\sizefn@info

```

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

```

345 \ifx\external@font\@empty
346 \else
347 \let\reserved@f\@empty
348 \fi
349 \fi
350 \fi
351 \reserved@f}}

```

`\lower@bound` `\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 the size against `\f@size`. If it is not equal `\f@size` it is not good and we have to recurse.

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

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

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

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

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

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

```
360 \else
```

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

```
362 \fi
```

```
363 \fi
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
377 (*trace)
```

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

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

```

380 \s@fct@}%
381 {\csname s@fct@#1\endcsname}%
382 </trace>
383 (-trace) \csname s@fct@#1\endcsname
384 }
385 </2ekernel | 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 (*2ekernel)
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 </2ekernel>
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 (*2ekernel)
400 \gdef\try@simples#1<#2>{%
401 \tryif@simple#2->\tryif@simple}

```

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

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

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

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

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

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

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

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

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

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

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

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

```
418 \else
```

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

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

```

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

```

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

```

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

```

If it's a range iterate also.

```

443 \reserved@f}

```

## 32.1 Sizefunctions

In the following we define some useful size functions.

`\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 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 family/series/shape combination, the optional argument (if present) is ignored. The font encoding scheme cannot be changed. Therefore, the first thing we do is to prepend the encoding scheme.

```

468 \DeclareSizeFunction{sub}{\sub@sfcnt\@font@warning}
469 \DeclareSizeFunction{ssub}{\sub@sfcnt\@font@info}

470 \def\sub@sfcnt#1{%
471 \edef\mandatory@arg{\f@encoding/\mandatory@arg}%

```

Next action is split the arg into its individual components and allow for a late font shape load.

```

472 \begingroup
473 \expandafter\split@name\mandatory@arg/\@nil
474 \try@load@fontshape
475 \endgroup

```

Then we record the current `\f@size` since it may get clobbered.

```

476 \let\f@user@size\f@size

```

Then we check whether this new combination is defined and give an error message if not. In this case we also switch to `\error@fontshape`.

```

477 \expandafter
478 \ifx\csname\mandatory@arg\endcsname\relax
479 \errmessage{No\space declaration\space for\space
480 shape\space \mandatory@arg}%
481 \error@fontshape
482 \else

```

Otherwise we warn the user about the substitution taking place.

```

483 #1{Font\space shape\space '\curr@fontshape'\space in\space
484 size\space <\f@size>\space not\space available\MessageBreak
485 Font\space shape\space '\mandatory@arg'\space tried\space
486 instead}%
487 \expandafter\split@name\mandatory@arg/\@nil
488 \fi

```

Then we restart the font specification scan by calling `\get@external@font`.

```

489 \edef\f@size{\f@user@size}%
490 \get@external@font

```

Finally `\do@subst@correction` is called to get the font name right.

```

491 \do@subst@correction
492 }

```

`\s@fct@subf` The `subf` size function allows substitution of another font. The mandatory argument is the external name of the font to be substituted, the optional argument a size scaling factor like in the default size function. The main difference to the default size function is the warning message.

```

493 \DeclareSizeFunction{subf}{\subf@sfcnt\@font@warning}
494 \DeclareSizeFunction{ssubf}{\subf@sfcnt\@font@info}

495 \def\subf@sfcnt#1{%
496 #1{Font\space shape\space '\curr@fontshape'\space in\space
497 size\space \f@size\space not\space available\MessageBreak
498 external\space font\space '\mandatory@arg'\space used}%
499 \empty@sfcnt#1%
500 }

```

`\s@fct@fixed` The `fixed` size function is for using a font at a different size than requested. A warning message is printed, and the external font to be used is taken from the mandatory argument. If an optional argument is present it is used as the ‘at’ size for the font. Otherwise the font is loaded at its design size.

```

501 \DeclareSizeFunction{fixed}{\fixed@sfcnt\@font@warning}
502 \DeclareSizeFunction{sfixed}{\fixed@sfcnt\@font@info}

503 \def\fixed@sfcnt#1{%
504 \ifx\optional@arg\@empty
505 \let\external@font\mandatory@arg
506 \else
507 \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
508 \fi
509 #1{External\space font\space '\external@font'\space loaded\space
510 for\space size\MessageBreak
511 <\f@size>}%
512 }
513 /2ekernel)

```

## File q

# ltfsscmp.dtx

This file contains the implementation of commands giving compatibility with the original ‘NFSS1’ release of the Font Selection Scheme.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

Version 1 of NFSS is obsolete now for about 20 years (and was “current” only for a short intermediate time) so with the 2015 release these internal interface commands are removed from the kernel and made available via `latexrelease` package so that backward compatibility remains ensured for very old documents.

```
1 (*latexrelease)
2 \IncludeInRelease{2015/01/01}{\new@fontshape}%
3 {NFSS version1 commands}%
4 \let\new@fontshape\@undefined
5 \let\warn@rel@i\@undefined
6 \let\scan@fontshape\@undefined
7 \let\scan@@fontshape\@undefined
8 \let\subst@fontshape\@undefined
9 \let\extra@def\@undefined
10 \let\default@mextra\@undefined
11 \let\preload@sizes\@undefined
12 \let\err@rel@i\@undefined
13 \let\newmathalphabet\@undefined
14 \let\newmathalphabet@\@undefined
15 \let\newmathalphabet@@\@undefined
16 \let@if@no@font@opt\@undefined
17 \let@no@font@optfalse\@undefined
18 \let\define@mathalphabet\@undefined
19 \let\define@mathgroup\@undefined
20 \let\addtoversion\@undefined
21 \EndIncludeInRelease
```

In older releases we provide the original definitions.

```
22 \IncludeInRelease{0000/00/00}{\new@fontshape}%
23 {NFSS version1 commands}%
```

`\new@fontshape` The interface is now `\DeclareFontShape`.

```
24 \gdef\new@fontshape#1#2#3#4{%
25 \warn@rel@i\new@fontshape\DeclareFontShape
26 \expandafter\scan@fontshape\@gobble#4<\@nil><<%
27 \DeclareFontShape U{#1}{#2}{#3}\reserved@f}%
28 \@onlypreamble\new@fontshape
```

`\warn@rel@i` The warning message used above.

```
29 \gdef\warn@rel@i#1#2{%
30 \@font@warning{*** NFSS release 1 command
31 \noexpand#1found\MessageBreak
32 *** Update by using release 2 command
```

```

33 \string#2.\MessageBreak
34 *** Recovery is probably possible}%
35 }%
36 \@onlypreamble\warn@rel@i

```

`\scan@fontshape` This will scan the old font shape definition syntax.

```

37 \gdef\scan@fontshape{%
38 \let\reserved@f\@empty
39 \let\reserved@e\@empty % holds last info
40 \scan@@fontshape
41 }%
42 \@onlypreamble\scan@fontshape

```

`\scan@@fontshape`

```

43 \gdef\scan@@fontshape#1>#2#3<{%
44 \ifx\@nil#1%
45 \edef\reserved@f{\reserved@f\reserved@e}%
46 \else
47 \def\reserved@b{#1}% nick names
48 \def\reserved@c{#3}%
49 \in@{ at}{#3}%
50 \ifin@
51 \in@{pt}{#3}% not a proof but a good chance
52 \ifin@

```

We grap also everything after pt and discard it if people have forgotten to place a percent sign there.

```

53 \def\reserved@a##1 at##2pt##3\@nil{%
54 \def\reserved@b{##2}%
55 \def\reserved@c{##1}%
56 }%
57 \reserved@a#3\@nil
58 \fi
59 \fi
60 \ifnum 0<0#2
61 \edef\reserved@d{subf*\reserved@c}%
62 \ifcase #2\or
63 \or
64 \else
65 \errmessage{*** What's this? NFSS release 0? ***}%
66 \fi
67 \else
68 \edef\reserved@d{#2\reserved@c}%
69 \fi
70 \ifx\reserved@d\reserved@e
71 \edef\reserved@f{\reserved@f<\reserved@b>}%
72 \else
73 \edef\reserved@f{\reserved@f\reserved@e<\reserved@b>}%add old info
74 \let\reserved@e\reserved@d
75 \fi
76 \expandafter\scan@@fontshape
77 \fi
78 }%
79 \@onlypreamble\scan@@fontshape

```



`\subst@fontshape` This is now also handled by the extend syntax of `\DeclareFontShape`.

```

80 \gdef\subst@fontshape#1#2#3#4#5#6{%
81 \warn@rel@i\subst@fontshape\DeclareFontShape
82 \DeclareFontShape{U}{#1}{#2}{#3}{<->sub*#4/#5/#6}{}}%
83 \@onlypreamble\subst@fontshape

```

`\extra@def` This was replaced by `\DeclareFontFamily`.

```

84 \gdef\extra@def#1#2#3{%
85 \warn@rel@i\extra@def\DeclareFontFamily
86 \DeclareFontFamily{U}{#1}{}}%
87 }%
88 \@onlypreamble\extra@def

```

`\default@mextra` The new name is `\DeclareFontEncodingDefaults` but in this case we don't feel comfortable with this either.

```

89 \gdef\default@mextra{%
90 \warn@rel@i\default@mextra\DeclareFontEncodingDefaults

```

We pick up the argument to `\default@mextra` implicitly as the second argument of `\DeclareFontEncodingDefaults`.

```

91 \DeclareFontEncodingDefaults\relax
92 }%
93 \@onlypreamble\default@mextra

```

`\preload@sizes` The new interface is `\DeclarePreloadSizes`.

```

94 \gdef\preload@sizes{%
95 \warn@rel@i\preload@sizes\DeclarePreloadSizes
96 \DeclarePreloadSizes U%
97 }%
98 \@onlypreamble\preload@sizes

```

`\err@rel@i` This macro is used in cases where emulation with NFSS2 features is not really possible.

```

99 \gdef\err@rel@i#1#2{%
100 \@latex@error{*** NFSS release 1 command \noexpand#1found%
101 ^^J*** Recovery not possible. Use \string#2}%
102 {The new release of NFSS doesn't support the
103 \noexpand#1command^^Jany longer.
104 Please upgrade your file to the syntax of NFSS
105 release 2^^Jusing the \noexpand#2command.}%

```

Let's die.

```

106 \batchmode\input.\relax
107 }%
108 \@onlypreamble\err@rel@i

```

`\newmathalphabet` `\newmathalphabet` is the old form.

```

\newmathalphabet@@
\newmathalphabet@@@
109 \gdef\newmathalphabet{%
110 \if@no@font@opt
111 \@latex@error{*** NFSS release 1 command
112 \noexpand\newmathalphabet found%
113 ^^J \space*** Automatic recovery not possible.%
114 ^^J \space*** TYPE H for Help%
115 }%

```

```

116 {Please look at the file usrguide.tex for hints on
117 how to resolve this problem.}%
118 \else
119 \warn@rel@i\newmathalphabet\DeclareMathAlphabet
120 \fi
121 \@ifstar\newmathalphabet@@
122 \newmathalphabet@@}%
123 \gdef\newmathalphabet@@#1{\DeclareMathAlphabet#1{U}{-}{-}}%
124 \gdef\newmathalphabet@@@#1#2#3#4{%
125 \DeclareMathAlphabet{#1}{U}{#2}{#3}{#4}}%
126 \@onlypreamble\newmathalphabet
127 \@onlypreamble\newmathalphabet@@
128 \@onlypreamble\newmathalphabet@@@

\if@no@font@opt
\@no@font@optfalse 129 \global\let\if@no@font@opt\iftrue
130 \gdef\@no@font@optfalse{\let\if@no@font@opt\iffalse}%

\define@mathalphabet This is a case where dying is best.
131 \gdef\define@mathalphabet{%
132 \err@rel@i\define@mathalphabet\DeclareMathAlphabet
133 }%
134 \@onlypreamble\define@mathalphabet

\define@mathgroup And here is another one
135 \gdef\define@mathgroup{%
136 \err@rel@i\define@mathgroup\DeclareSymbolFont
137 }%
138 \@onlypreamble\define@mathgroup

\addtoversion \addtoversion is the old form.
139 \def\addtoversion#1#2{%
140 \warn@rel@i\addtoversion\SetMathAlphabet
141 \SetMathAlphabet#2{#1}{U}}%
142 \@onlypreamble\addtoversion

 Finishing off this huge \IncludeInRelease argument:
143 \EndIncludeInRelease
144 </latexrelease>

```

## File r

# lftssdcl.dtx

This file contains the main implementation of the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

**Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

## 33 Interface Commands

`\in@` `\@in` is a utility macro with two arguments. It determines whether its first argument occurs in its second and sets the switch `\ifin@` accordingly. The first argument may not contain braces nor # (more precisely, tokens of category code 1, 2, or 6).

```
1 (*2kernel)
2 \def\in@#1#2%
3 {%
4 \begingroup
5 \def\in@@##1#1{%
6 \toks@\expandafter{\in@@#2{}}#1%
7 \edef\in@@{\the\toks@}%
8 \expandafter\endgroup
9 \ifx\in@@\@empty
10 \in@false
11 \else
12 \in@true
13 \fi
14 }
15 \newif\ifin@
```

Before the `\begin{document}` command several *math versions* and *math alphabet identifiers* may be declared. In principle, there should be exactly one family/series/shape combination be declared for each version/alphabet pair. But we want to allow for defaults as well for automagical filling of holes.

While building the tables for math alphabet identifiers and math versions we keep several lists:

- the list of all math versions, `\version@list`, each entry prefixed by the control sequence `\version@elt`, i.e. this list has the following form

```
\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>\}}.
```



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 \begingroup
58 \escapechar\m@ne
59 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
60 \globaldefs\@ne \math@fonts
61 \endgroup
62 \init@restore@version
63 \xdef#1{\noexpand\use@mathgroup\noexpand#2%
64 {\number\csname c@mv@\math@version\endcsname}}%
65 \global\advance\csname c@mv@\math@version\endcsname\@ne
66 \else
67 \let#1\relax
68 \@latex@error{Too many math alphabets used in
69 version \math@version}%
70 \@eha
71 \fi
72 \else \expandafter\non@alpherr\fi
73 #1{#4}%
74 }%
75 }
76 </2ekernel | latexrelease>
77 <latexrelease>\EndIncludeInRelease
78 <latexrelease>\IncludeInRelease{0000/00/00}
79 <latexrelease> {\select@group}{\select@group}%
80 <latexrelease>\def\select@group#1#2#3#4{%
81 <latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
82 <latexrelease> {%
83 <latexrelease> \ifmmode
84 <latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\sixt@n
85 <latexrelease> \begingroup
86 <latexrelease> \escapechar\m@ne
87 <latexrelease> \getanddefine@fonts
88 <latexrelease> {\csname c@mv@\math@version\endcsname}#3%
89 <latexrelease> \globaldefs\@ne \math@fonts
90 <latexrelease> \endgroup
91 <latexrelease> \init@restore@version
92 <latexrelease> \xdef#1{\noexpand\use@mathgroup\noexpand#2%
93 <latexrelease> {\number\csname c@mv@\math@version\endcsname}}%
94 <latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
95 <latexrelease> \else
96 <latexrelease> \let#1\relax
97 <latexrelease> \@latex@error{Too many math alphabets used in
98 <latexrelease> version \math@version}%
99 <latexrelease> \@eha
100 <latexrelease> \fi
101 <latexrelease> \else \expandafter\non@alpherr\fi

```

```

102 <latexrelease> #1{#4}%
103 <latexrelease> }%
104 <latexrelease>}
105 <latexrelease>\EndIncludeInRelease
106 <*/2ekernel)

107 \@onlypreamble\restore@mathversion

\init@restore@version
108 \def\init@restore@version{%
109 \global\let\init@restore@version\relax
110 \xdef\restore@mathversion
111 {\expandafter\noexpand\csname mv@\math@version\endcsname
112 \global\csname c@mv@\math@version\endcsname
113 \number\csname c@mv@\math@version\endcsname\relax}%
114 \aftergroup\dorestore@version
115 }
116 \@onlypreamble\init@restore@version

\non@alpherr
117 \gdef\non@alpherr#1{\@latex@error{%
The command here will have a space at the end of its name, so we make sure not
to insert an extra one.
118 \string#1allowed only in math mode}\@ehd}

\dorestore@version
119 \def\dorestore@version
120 {\ifmmode
121 \aftergroup\dorestore@version
122 \else
123 \gdef\init@restore@version{%
124 \global\let\init@restore@version\relax
125 \xdef\restore@mathversion
126 {\expandafter\noexpand\csname mv@\math@version\endcsname
127 \global\csname c@mv@\math@version\endcsname
128 \number\csname c@mv@\math@version\endcsname\relax}%
129 \aftergroup\dorestore@version
130 }%
131 \begingroup
132 \let\getanddefine@fonts@gobbletwo
133 \restore@mathversion
134 \endgroup
135 \fi}%
136 \@onlypreamble\dorestore@version

\document@select@group We surround \select@group with braces so that functions using it can be used
directly after _ or ^.
137 </2ekernel)
138 <latexrelease>\IncludeInRelease{2015/01/01}
139 <latexrelease> {\document@select@group}{\document@select@group}%
140 <*/2ekernel | latexrelease)
141 \def\document@select@group#1#2#3#4{%
142 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi

```

```

143 {%
144 \ifmmode
145 \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
146 \begingroup
147 \escapechar\m@ne
148 \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
149 \globaldefs\@ne \math@fonts
150 \endgroup
151 \expandafter\extract@alph@from@version
152 \csname mv@\math@version\expandafter\endcsname
153 \expandafter{\number\csname
154 c@mv@\math@version\endcsname}%
155 #1%
156 \global\advance\csname c@mv@\math@version\endcsname\@ne
157 \else
158 \let#1\relax
159 \@latex@error{Too many math alphabets used
160 in version \math@version}%
161 \@eha
162 \fi
163 \else \expandafter\non@alpherr\fi
164 #1{#4}%
165 }%
166 }
167 </2ekernel | latexrelease>
168 <latexrelease>\EndIncludeInRelease
169 <latexrelease>\IncludeInRelease{0000/00/00}
170 <latexrelease> {\document@select@group}{\document@select@group}%
171 <latexrelease>\def\document@select@group#1#2#3#4{%
172 <latexrelease> \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
173 <latexrelease> {%
174 <latexrelease> \ifmmode
175 <latexrelease> \ifnum\csname c@mv@\math@version\endcsname<\sist@n
176 <latexrelease> \begingroup
177 <latexrelease> \escapechar\m@ne
178 <latexrelease> \getanddefine@fonts
179 <latexrelease> {\csname c@mv@\math@version\endcsname}#3%
180 <latexrelease> \globaldefs\@ne \math@fonts
181 <latexrelease> \endgroup
182 <latexrelease> \expandafter\extract@alph@from@version
183 <latexrelease> \csname mv@\math@version\expandafter\endcsname
184 <latexrelease> \expandafter{\number\csname
185 <latexrelease> c@mv@\math@version\endcsname}%
186 <latexrelease> #1%
187 <latexrelease> \global\advance\csname c@mv@\math@version\endcsname\@ne
188 <latexrelease> \else
189 <latexrelease> \let#1\relax
190 <latexrelease> \@latex@error{Too many math alphabets used
191 <latexrelease> in version \math@version}%
192 <latexrelease> \@eha
193 <latexrelease> \fi
194 <latexrelease> \else \expandafter\non@alpherr\fi
195 <latexrelease> #1{#4}%
196 <latexrelease> }%

```



```

197 (latexrelease)}
198 (latexrelease)\EndIncludeInRelease
199 (*2ekernel)

```

`\process@table`

```

200 \def\process@table{%
201 \def\cdp@elt##1##2##3##4{%
202 \@font@info{Checking defaults for
203 ##1/##2/##3/##4}%
204 \expandafter
205 \ifx\csname##1/##2/##3/##4\endcsname\relax

```

Grouping is important for two reasons, first `\cdp@elt` will get redefined if `\Declare...` functions are executed within the external `.fd` file and secondly `\try@load@fontshape` changes a lot of catcodes without surrounding itself with a group.

```

206 \begingroup
207 \def\fontencoding{##1}\def\fontfamily{##2}%
208 \try@load@fontshape
209 \endgroup
210 \fi
211 \expandafter
212 \ifx\csname##1/##2/##3/##4\endcsname\relax
213 \@latex@error{This NFSS system isn't set up properly}%
214 {For encoding scheme ##1 the defaults
215 ##2/##3/##4 do not form a valid font shape}%
216 \else
217 \@font@info{... okay}%
218 \fi}%
219 \cdp@list

```

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

```

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

```

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

```

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

```

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

```

237 \fontencoding{\encodingdefault}%

```

```

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

244 %\@onlypreamble\set@mathradical

```

`\DeclareMathVersion`

```

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

```

`\new@mathversion`

```

248 \def\new@mathversion#1{%
249 \expandafter\in@\expandafter#1\expandafter{\version@list}%
250 \ifin@
251 \@font@info{Redeclaring 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 \@tempswafalse
282 \edef\reserved@b{#2}%
283 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
284 \ifx\reserved@b\reserved@c \@tempswatruer\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 \@tempwattrue\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\@alphabet@error \@alphabet@error}%
557 \gdef#4{\@alphabet@error #5}% fake things :-)
Then call the internal setting routine again:
558 \SetMathAlphabet@{#1}{#2}{#3}#4#5%
559 \else
560 \@latex@error{Command ‘\string#5’ not defined as a
561 math alphabet}%
562 {Use \noexpand\DeclareMathAlphabet to define it.}%
563 \fi
564 \fi
565 \else
566 \@latex@error{Math version ‘\expandafter\@gobblefour\string#1’
567 is not
568 defined}{You probably misspelled the name of the math
569 version.^^JOr you have to specify an additional package.}%
570 \fi
571 }
572 \@onlypreamble\SetMathAlphabet@

```

`\DeclareMathAlphabet` could do with more checks like allowing single number in #4 lowercase in #4 etc

```

573 \def\DeclareMathAccent#1#2#3#4{%
574 \expandafter\in@\csname sym#3\expandafter\endcsname
575 \expandafter{\group@list}%
576 \ifin@
577 \begingroup
578 \count\z@=#4\relax
579 \count\tw@\count\z@
580 \divide\count\z@\sist@@n
581 \count@\count\z@
582 \multiply\count@\sist@@n
583 \advance\count\tw@-\count@
584 \if\relax\noexpand#1% is command?
585 \edef\reserved@a{\noexpand\in@
586 {\expandafter\@gobble\string\mathaccent}{\meaning#1}}%
587 \reserved@a
588 \ifin@
589 \expandafter\set@mathaccent
590 \csname sym#3\endcsname#1#2%
591 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
592 \@font@info{Redeclaring math accent \string#1}%
593 \else
594 \expandafter\ifx
595 \csname\expandafter\@gobble\string#1\endcsname
596 \relax
597 \expandafter\set@mathaccent
598 \csname sym#3\endcsname#1#2%
599 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
600 \else
601 \@latex@error{Command ‘\string#1’ already defined}\@eha
602 \fi
603 \fi
604 \else
605 \@latex@error{Not a command name: ‘\noexpand#1’}\@eha

```

```

606 \fi
607 \endgroup
608 \else
609 \@latex@error{Symbol font ‘#3’ is not defined}\@eha
610 \fi
611 }
612 \@onlypreamble\DeclareMathAccent

```

\set@mathaccent

```

613 \def\set@mathaccent#1#2#3#4{%
614 \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}}
615 \@onlypreamble\set@mathaccent

```

\DeclareMathSymbol

```

616 \def\DeclareMathSymbol#1#2#3#4{%
617 \expandafter\in@\csname sym#3\expandafter\endcsname
618 \expandafter{\group@list}%
619 \ifin@
620 \begingroup
621 \count\z@=#4\relax
622 \count\tw@\count\z@
623 \divide\count\z@\sist@@n
624 \count@\count\z@
625 \multiply\count@\sist@@n
626 \advance\count\tw@-\count@
627 \if\relax\noexpand#1% is command?
628 \edef\reserved@a
629 {\noexpand\in@{\expandafter@gobble\string\mathchar}{\meaning#1}}%
630 \reserved@a
631 \ifin@
632 \expandafter\set@mathsymbol
633 \csname sym#3\endcsname#1#2%,
634 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%,
635 \@font@info{Redefining math symbol \string#1}%
636 \else
637 \expandafter\ifx
638 \csname\expandafter@gobble\string#1\endcsname
639 \relax
640 \expandafter\set@mathsymbol
641 \csname sym#3\endcsname#1#2%,
642 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%,
643 \else
644 \@latex@error{Command ‘\string#1’ already defined}\@eha
645 \fi
646 \fi
647 \else
648 \expandafter\set@mathchar
649 \csname sym#3\endcsname#1#2%,
650 {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%,
651 \fi
652 \endgroup
653 \else
654 \@latex@error{Symbol font ‘#3’ is not defined}\@eha
655 \fi

```

```

656 }
657 \@onlypreamble\DeclareMathSymbol

\set@mathchar
658 \def\set@mathchar#1#2#3#4{%
659 \global\mathcode'#2="\mathchar@type#3\hexnumber@#1#4\relax}
660 \@onlypreamble\set@mathchar

\set@mathsymbol
661 \def\set@mathsymbol#1#2#3#4{%
662 \global\mathchardef#2"\mathchar@type#3\hexnumber@#1#4\relax}
663 \@onlypreamble\set@mathsymbol

664 %\def\mathsymbol#1#2#3{%
665 % \@tempcnta=#3\relax
666 % \@tempcntb\@tempcnta
667 % \divide\@tempcnta\sixt@@n
668 % \count@\@tempcnta
669 % \multiply\count@\sixt@@n
670 % \advance\@tempcntb-\count@
671 % \mathchar"\mathchar@type#1\hexnumber@#2%
672 % \hexnumber@\@tempcnta\hexnumber@\@tempcntb\relax}
673 %
674 %\def\DeclareMathAlphabetCharacter#1#2#3{%
675 % \DeclareMathSymbol{#1}7{#2}{#3}}

```

`\DeclareMathDelimiter`

```

676 \def\DeclareMathDelimiter#1{%
677 \if\relax\noexpand#1%
678 \expandafter\@DeclareMathDelimiter
679 \else
680 \expandafter\@xxDeclareMathDelimiter
681 \fi
682 #1}
683 \@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.

```

684 \def\@xxDeclareMathDelimiter#1#2#3#4{%

```

7 is the default value returned in the case that `\mathchar@type` is passed something unexpected, like a math symbol font name. We locally move `\mathalpha` out of the way so if you use that the right branch is taken. This will still fail if an explicit number 7 is used!

```

685 \begingroup
686 \let\mathalpha\mathord
687 \ifnum7=\mathchar@type{#2}%
688 \endgroup

```

If this branch is taken we have old syntax (5 arguments).

```

689 \expandafter\@firstofone
690 \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.

```
691 \endgroup
692 \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.

```
693 \expandafter\@firstoftwo
694 \fi
695 {\@xDeclareMathDelimiter#1}{#2}{#3}{#4}}
696 \@onlypreamble\@xxDeclareMathDelimiter
```

`\@DeclareMathDelimiter`

```
697 \def\@DeclareMathDelimiter#1#2#3#4#5#6{%
698 \expandafter\in@\csname sym#3\expandafter\endcsname
699 \expandafter{\group@list}%
700 \ifin@
701 \expandafter\in@\csname sym#5\expandafter\endcsname
702 \expandafter{\group@list}%
703 \ifin@
704 \begingroup
705 \count\z@=#4\relax
706 \count\tw@\count\z@
707 \divide\count\z@\sixt@n
708 \count@\count\z@
709 \multiply\count@\sixt@n
710 \advance\count\tw@-\count@
711 \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
712 %
713 \count\z@=#6\relax
714 \count\tw@\count\z@
715 \divide\count\z@\sixt@n
716 \count@\count\z@
717 \multiply\count@\sixt@n
718 \advance\count\tw@-\count@
719 \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
720 %
721 \edef\reserved@a{\noexpand\in@
722 {\expandafter\@gobble\string\delimiter}{\meaning#1}}%
723 \reserved@a
724 \ifin@
725 \expandafter\set@mathdelimiter
726 \csname sym#3\expandafter\endcsname
727 \csname sym#5\endcsname#1#2%
728 \reserved@c\reserved@d
729 \@font@info{Redeclaring math delimiter \string#1}%
730 \else
731 \expandafter\ifx
732 \csname\expandafter\@gobble\string#1\endcsname
733 \relax
734 \expandafter\set@mathdelimiter
735 \csname sym#3\expandafter\endcsname
736 \csname sym#5\endcsname#1#2%
```

```

737 \reserved@c\reserved@d
738 \else
739 \@latex@error{Command ‘\string#1’ already defined}\@eha
740 \fi
741 \fi
742 \endgroup
743 \else
744 \@latex@error{Symbol font ‘#5’ is not defined}\@eha
745 \fi
746 \else
747 \@latex@error{Symbol font ‘#3’ is not defined}\@eha
748 \fi
749 }
750 \@onlypreamble\@DeclareMathDelimiter

```

`\@xDeclareMathDelimiter`

```

751 \def\@xDeclareMathDelimiter#1#2#3#4#5{%
752 \expandafter\in@\csname sym#2\expandafter\endcsname
753 \expandafter{\group@list}%
754 \ifin@
755 \expandafter\in@\csname sym#4\expandafter\endcsname
756 \expandafter{\group@list}%
757 \ifin@
758 \begingroup
759 \count\z@=#3\relax
760 \count\tw@\count\z@
761 \divide\count\z@\sixt@n
762 \count@\count\z@
763 \multiply\count@\sixt@n
764 \advance\count\tw@-\count@
765 \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
766 %
767 \count\z@=#5\relax
768 \count\tw@\count\z@
769 \divide\count\z@\sixt@n
770 \count@\count\z@
771 \multiply\count@\sixt@n
772 \advance\count\tw@-\count@
773 \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
774 \expandafter\set@mathdelimiter
775 \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
776 \reserved@c\reserved@d
777 \endgroup
778 \else
779 \@latex@error{Symbol font ‘#4’ is not defined}\@eha
780 \fi
781 \else
782 \@latex@error{Symbol font ‘#2’ is not defined}\@eha
783 \fi
784 }
785 \@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)

```
786 \def\set@mathdelimiter#1#2#3#4#5#6{%
787 \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
788 \hexnumber@#2#6 }}
789 \@onlypreamble\set@mathdelimiter
```

\set@@mathdelimiter

```
790 \def\set@@mathdelimiter#1#2#3#4#5{%
791 \global\delcode'#3="\hexnumber@#1#4\hexnumber@#2#5\relax}
792 \@onlypreamble\set@@mathdelimiter
```

\DeclareMathRadical

```
793 \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!

```
794 \expandafter\ifx
795 \csname\expandafter\@gobble\string#1\endcsname
796 \relax
797 \let#1\radical
798 \fi
799 \edef\reserved@a{\noexpand\in@
800 {\expandafter\@gobble\string\radical}{\meaning#1}}%
801 \reserved@a
802 \ifin@
803 \expandafter\in@\csname sym#2\expandafter\endcsname
804 \expandafter{\group@list}%
805 \ifin@
806 \expandafter\in@\csname sym#4\expandafter\endcsname
807 \expandafter{\group@list}%
808 \ifin@
809 \begin@group
810 \count\z@=#3\relax
811 \count\tw@\count\z@
812 \divide\count\z@\sixt@@n
813 \count@\count\z@
814 \multiply\count@\sixt@@n
815 \advance\count\tw@-\count@
816 \edef\reserved@c{%
817 \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
818 \count\z@=#5\relax
819 \count\tw@\count\z@
820 \divide\count\z@\sixt@@n
821 \count@\count\z@
822 \multiply\count@\sixt@@n
823 \advance\count\tw@-\count@
824 \edef\reserved@d{%
825 \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
```

Coded inline instead of using \set@mathradical

```
826 % \expandafter\set@mathradical
827 % \csname sym#2\expandafter\endcsname
828 % \csname sym#4\endcsname#1%
829 % \reserved@c\reserved@d
```

```

830 \xdef#1{\radical"\expandafter\hexnumber@
831 \csname sym#2\endcsname\reserved@c
832 \expandafter\hexnumber@
833 \csname sym#4\endcsname\reserved@d
834 \relax}%
835 \endgroup
836 \else
837 \@latex@error{Symbol font ‘#4’ is not defined}\@eha
838 \fi
839 \else
840 \@latex@error{Symbol font ‘#2’ is not defined}\@eha
841 \fi
842 \else
843 \@latex@error{Command ‘\string#1’ already defined}\@eha
844 \fi
845 }
846 \@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
847 \let\mathalpha\relax

```

\mathchar@type

```

848 \def\mathchar@type#1{%
849 \ifodd 2#11 #1\else % is this non-negative number?
850 \ifx#1\mathord 0\else
851 \ifx#1\mathop 1\else
852 \ifx#1\mathbin 2\else
853 \ifx#1\mathrel 3\else
854 \ifx#1\mathopen 4\else
855 \ifx#1\mathclose 5\else
856 \ifx#1\mathpunct 6\else
857 7% % anything else is variable ord
858 \fi
859 \fi
860 \fi
861 \fi
862 \fi
863 \fi
864 \fi
865 \fi}
866 \@onlypreamble\mathchar@type

```

\DeclareSymbolFontAlphabet

```

867 \def\DeclareSymbolFontAlphabet#1#2{%
868 \expandafter\DeclareSymbolFontAlphabet@
869 \csname \expandafter@gobble\string#1\space\endcsname{#2}#1}
870 \@onlypreamble\DeclareSymbolFontAlphabet

```

\DeclareSymbolFontAlphabet@

```

871 \def\DeclareSymbolFontAlphabet@#1#2#3{%

```

We use the switch `\if@tempswa` to decide if we can declare this symbol font alphabet.

```
872 \@tempswatrue
```

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

```
873 \expandafter\in@\csname sym#2\expandafter\endcsname
874 \expandafter{\group@list}%
875 \ifin@
```

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

```
876 \expandafter\in@\expandafter#1\expandafter{\alpha@list}%
877 \ifin@
```

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

```
878 \@font@info{Redeclaring math alphabet \string#3}%
879 \toks@{}%
880 \def\alpha@elt##1##2##3{%
881 \ifx##1#1\else\addto@hook\toks@{\alpha@elt##1##2##3}\fi}%
882 \alpha@list
883 \xdef\alpha@list{\the\toks@}%
```

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

```
884 \def\version@elt##1{%
885 \begingroup
886 \toks@{}%
887 \def\getanddefine@fonts####1####2{%
888 \addto@hook\toks@{\getanddefine@fonts####1####2}}%
889 \def\install@mathalphabet####1####2{%
890 \ifx####1#1\else
891 \addto@hook\toks@{\install@mathalphabet
892 ####1{####2}}\fi}%
893 ##1%
894 \xdef##1{\the\toks@}%
895 \endgroup
896 }%
897 \version@list
898 \else
```

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

```
899 \expandafter\ifx
900 \csname\expandafter\@gobble\string#1\space\endcsname
901 \relax
```

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

```
902 \else
903 \edef\reserved@a{%
904 \noexpand\in@{\string\use@mathgroup}{\meaning#1}}%
905 \reserved@a
906 \ifin@
907 \@font@info{Redeclaring math alphabet \string#3}%
908 \else
```

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

```
909 \@tempwafalse
```



```

910 \@latex@error{Command ‘\string#3’ already defined}\@eha
911 \fi
912 \fi
913 \fi
914 \else

```

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

```

915 \@tempswafalse
916 \@latex@error{Unknown symbol font ‘#2’}\@eha
917 \fi
918 \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.

```

919 \def\group@elt##1##2{%
920 \expandafter\ifx\csname sym#2\endcsname##1%
921 \expandafter\reserved@a\string##2\@nil
922 \fi}%
923 \def\reserved@a##1##2/##3\@nil{%
924 \def\reserved@a{##2}}%
925 \group@list
926 \toks@{\relax\ifmode \else \non@alpherr#1\fi}%
927 \edef#1{\the\toks@
928 \noexpand\use@mathgroup
929 \expandafter\noexpand\csname M@\reserved@a\endcsname
930 \csname sym#2\endcsname}%
931 \def#3{\protect#1}%
932 \fi
933 }
934 \@onlypreamble\DeclareSymbolFontAlphabet@
935 </2kernel>

```

## File s

# ltfssini.dtx

This file contains the top level L<sup>A</sup>T<sub>E</sub>X interface to the font selection scheme commands. See other parts of the L<sup>A</sup>T<sub>E</sub>X distribution, or *The L<sup>A</sup>T<sub>E</sub>X Companion* for higher level documentation of these commands.

## 34 NFSS Initialisation

Finally, there are six commands that are to be used in L<sup>A</sup>T<sub>E</sub>X and that we will therefore protect against expansion at the wrong point: `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize`, `\selectfont`, and `\mathversion`.

```
1 <*2ekernel>
```

### 34.1 Providing math *versions*

L<sup>A</sup>T<sub>E</sub>X provides two *versions*. We call them normal and bold, respectively.

```
2 \DeclareMathVersion{normal}
3 \DeclareMathVersion{bold}
```

Now we define the standard font change commands. We don't allow the use of `\rmfamily` etc. in math mode.

First the changes to another *family*:

```
4 \DeclareRobustCommand\rmfamily
5 {\not@math@alphabet\rmfamily\mathrm
6 \fontfamily\rmdefault\selectfont}
7 \DeclareRobustCommand\sffamily
8 {\not@math@alphabet\sffamily\mathsf
9 \fontfamily\sfdefault\selectfont}
10 \DeclareRobustCommand\ttfamily
11 {\not@math@alphabet\ttfamily\mathtt
12 \fontfamily\ttdefault\selectfont}
```

Then the commands changing the *series*:

```
13 \DeclareRobustCommand\bfseries
14 {\not@math@alphabet\bfseries\mathbf
15 \fontseries\bfdefault\selectfont}
16 \DeclareRobustCommand\mdseries
17 {\not@math@alphabet\mdseries\relax
18 \fontseries\mddefault\selectfont}
19 \DeclareRobustCommand\upshape
20 {\not@math@alphabet\upshape\relax
21 \fontshape\updefault\selectfont}
```

Then the commands changing the *shape*:

```
22 \DeclareRobustCommand\slshape
23 {\not@math@alphabet\slshape\relax
24 \fontshape\sldefault\selectfont}
25 \DeclareRobustCommand\scshape
26 {\not@math@alphabet\scshape\relax
27 \fontshape\scdefault\selectfont}
```

```

28 \DeclareRobustCommand\itshape
29 {\not@math@alphabet\itshape\mathit
30 \fontshape\itdefault\selectfont}

```

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

```

\em
\eminersshape
31 </2kernel>
32 <latexrelease>\IncludeInRelease{2015/01/01}{\eminersshape}{\eminersshape}%
33 <*2kernel | latexrelease>
34 \DeclareRobustCommand\em
35 {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
36 \eminersshape \else \itshape \fi}%
37 \def\eminersshape{\upshape}%
38 </2kernel | latexrelease>
39 <latexrelease>\EndIncludeInRelease
40 <latexrelease>\IncludeInRelease{0000/00/00}{\eminersshape}{\eminersshape}%
41 <latexrelease>\DeclareRobustCommand\em
42 <latexrelease> {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
43 <latexrelease> \upshape \else \itshape \fi}%
44 <latexrelease>\let\eminersshape\@undefined
45 <latexrelease>\EndIncludeInRelease
46 <*2kernel>

```

`\not@math@alphabet` This function generates an error message when it is called in math mode. The same function should be defined in `newlfont.sty`.

```

47 \def\not@math@alphabet#1#2{%
48 \relax
49 \ifmmode
50 \@latex@error{Command \noexpand#1invalid in math mode}%
51 {%
52 Please
53 \ifx#2\relax
54 define a new math alphabet^^J%
55 if you want to use a special font in math mode%
56 \else

```

We have to a `\noexpand` below to prevent expansion of `#2`. In case of `#1` we can omit this (due to the current definition of robust commands since they do come out right there :-).

```

57 use the math alphabet \noexpand#2instead of
58 the #1command%
59 \fi
60 .
61 }%
62 \fi}

```

Finally we provide two abbreviations to switch to the L<sup>A</sup>T<sub>E</sub>X *versions*.

```

63 \def\boldmath{\@nomath\boldmath
64 \mathversion{bold}}
65 \def\unboldmath{\@nomath\unboldmath
66 \mathversion{normal}}

```

Here we switch to the default math version by defining the internal macro `\math@version`. We dare not to call `\mathversion` at this place because this would call `\glb@settings`.

```
67 \def\math@version{normal}
```

## 34.2 Miscellaneous

`\newfont` We start by defining a few macros that are part of standard L<sup>A</sup>T<sub>E</sub>X's user interface.  
`\symbol` The use of these functions is not encouraged, but they will allow to process older documents without changes to the source.

```
68 \def\newfont#1#2{\ifdefinable#1{\font#1=#2\relax}}
69 \def\symbolsymbol#1{\char #1\relax}
```

`\@setfontsize` This abbreviation is used by L<sup>A</sup>T<sub>E</sub>X's user level size changing commands, such as  
`\@setsize` `\large`.

```
70 \def\@setfontsize#1#2#3{\@nomath#1%
```

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

```
71 \ifx\protect\@typeset@protect
72 \let\@currsize#1%
73 \fi
74 \fontsize{#2}{#3}\selectfont}
```

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

```
75 *compat)
76 \def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
77 \</compat)
```

`\oldstylenums` This macro implements old style numerals but only works if we assume that the standard math fonts are used. Thus it needs changing in case other math encodings are used.

```
78 \def\oldstylenums#1{%
79 \begingroup
```

Provide spacing using the interword space of the current font.

```
80 \spaceskip\fontdimen\tw@\font
```

Then switch to the math italic font. We don't change the current value of `\f@series` which means that you can use bold numerals if `\bfseries` is in force. As family we use `\rmdefault` which means that this only works if there exist an OML encoded version of that font or rather a corresponding `.fd` file (which is the case for standard L<sup>A</sup>T<sub>E</sub>X fonts even though they only contain substitutions).

```
81 \usefont{OML}{\rmdefault}{\f@series}{it}%
82 \mathgroup\symbols #1%
83 \endgroup
84 }
```

`\hexnumber@` To set up L<sup>A</sup>T<sub>E</sub>X's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple `\ifcase`.

```
85 \def\hexnumber@#1{\ifcase\number#1
```

```

86 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
87 9\or A\or B\or C\or D\or E\or F\fi}

```

`\nfss@text` In its simplest form `\nfss@text` is an `\mbox`. This will produce unbreakable text outside math and inside math you will get text with the same fonts as outside. The only drawback is that such item won't change sizes in subscripts. But this behavior can be easily changed. With the `amstex` style option one will get a sub style called `amstext` which will redefine the `\nfss@text` macro to produce correct text in all sizes.

We have to use `\def` instead of the shorter `\let` since `\mbox` is undefined when we reach this point.

```
88 \def\nfss@text#1{\mbox{#1}}
```

`\copyright` The definition of `\copyright` was changed so that it works in other type styles, and to make it robust. We leave the family untouched so that the copyright notice will come out differently if a different font family is in use. This command is commented out, since it is now defined in `ltoutenc.dtx`.

```

89 %\DeclareRobustCommand\copyright
90 % {\oalign{\hfil
91 % \raise.07ex\hbox{\mdseries\upshape c}\hfil\cr
92 % \mathhexbox20D}}

```

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

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

```

93 \DeclareRobustCommand\normalfont
94 {\usefont\encodingdefault
95 \familydefault
96 \seriesdefault
97 \shapedefault
98 \relax}
99 \let\reset@font\normalfont

```

We left out the special L<sup>A</sup>T<sub>E</sub>X fonts which are not automatically included in the base version of the font selection since these fonts contain only a few characters which are also included in the AMS fonts so anybody who is using these fonts doesn't need them. But for compatibility reasons we will define these symbols.

```

100 \def\not@base#1{\@latex@error
101 {Command \noexpand#1not provided in base LaTeX2e}%
102 {Load the latexsym or the amsfonts package to
103 define this symbol}}
104 \def\mho{\not@base\mho}
105 \def\Join{\not@base\Join}
106 \def\Box{\not@base\Box}
107 \def\Diamond{\not@base\Diamond}
108 \def\leadsto{\not@base\leadsto}
109 \def\squsubset{\not@base\squsubset}
110 \def\sqsupset{\not@base\sqsupset}
111 \def\lhd{\not@base\lhd}

```



```

152 ^^J%
153 =====}%
154 \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
155 }
156 {\input{fontmath.ltx}}
157 \let\@addtofilelist\@gobble

```

Then we preload several fonts. This file might be customized *without* changing the behavior of the format (i.e. necessary font definitions will be loaded at runtime if they are not preloaded). This is done in the file `preload.ltx`.

```

158 \InputIfFileExists{preload.cfg}
159 {\typeout{=====^^J%
160 ^^J%
161 Local config file preload.cfg used^^J%
162 ^^J%
163 =====}%
164 \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
165 }
166 {\input{preload.ltx}}
167 \let\@addtofilelist\@gobble

```

```

\@acci We also save the values of some accents in \@acci, \@accii and \@acciii so they
\@accii can be restored by a minipage inside a tabbing environment.
\@acciii 168 \let\@acci\' \let\@accii\' \let\@acciii\=

```

```

\cal Here were the two old <alphabet identifiers>.
\mit 169 </2kernel>

```

# File t

## fontdef.dtx

### 35 Introduction

This file is used to generate the files `fonttext.ltx` (text font declarations) and `fontmath.ltx` (math font declarations), which are used during the format generation. It contains the declaration of the standard text encodings used at the site as well as a minimal subset of font shape groups that NFSS will look at to ensure that the specified encodings are valid.

The math part contains the setup for math encodings as well as the default math symbol declarations that belong to the encoding.

It is possible to change this setup (by using other fonts, or defaults) without losing the ability to process documents written at other sites. Portability in this sense means that a document will compile without errors. It does not mean, however, that identical output will be produced. For this it is necessary that the distributed setup is used at both installations.

### 36 Customization

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

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

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

**Warning:** please note that we don't support customised  $\text{\LaTeX}$  versions. Thus, before sending in a bug report please try your test file with a  $\text{\LaTeX}$  format which is not customised and send in the log from that version (unless the problem goes away).

Please note: the following standard encodings have to be defined in all local variants of `font...cfg` to guarantee that all  $\text{\LaTeX}$  installations behave in the same way.

|     |                                                   |
|-----|---------------------------------------------------|
| T1  | Cork $\text{\TeX}$ text encoding                  |
| OT1 | old $\text{\TeX}$ text encoding                   |
| U   | unknown encoding                                  |
| OML | old $\text{\TeX}$ math letters encoding           |
| OMS | old $\text{\TeX}$ math symbols encoding           |
| OMX | old $\text{\TeX}$ math extension symbols encoding |



Notice that some of these encodings are ‘old’ in the sense that we hope that they will be superseded soon by encoding standards defined by the T<sub>E</sub>X user community. Therefore this set of default encodings may change in the future.

The first candidate is OT1 which will soon be replaced by T1, the official T<sub>E</sub>X text encoding.

**Warning:** If you add additional encodings to this file there is no guarantee any longer that files processable at your installation will also be processable at other installations. Thus, if you make use of such an encoding in your document, e.g. if you intend to typeset in Cyrillic (OT2 encoding), you need to specify this encoding in the preamble of your document prior to sending it to another installation. Once the encoding is specified in that place in your document, the document is processable at all L<sup>A</sup>T<sub>E</sub>X installations (provided they have suitable fonts installed).

For this reason we suggest that you define a short package file that sets up an additional encoding used at your site (rather than putting the encoding into this file) since this package can easily be shipped with your document.

## 37 The docstrip modules

The following modules are used to direct `docstrip` in generating external files:

|                      |                                                |
|----------------------|------------------------------------------------|
| <code>driver</code>  | produce a documentation driver file            |
| <code>text</code>    | produce the file <code>fonttext.ltx</code>     |
| <code>math</code>    | produce the file <code>fontmath.ltx</code>     |
| <code>cfgtext</code> | produce a dummy <code>fonttext.cfg</code> file |
| <code>cfgmath</code> | produce a dummy <code>fontmath.cfg</code> file |

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

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

## 38 A driver for this document

The next bit of code contains the documentation driver file for T<sub>E</sub>X, i.e. the file that will produce the documentation you are currently reading. It will be extracted from this file by the DOCSTRIP program.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 \GetFileInfo{fontdef.dtx}
4 \begin{document}
5 \DocInput{fontdef.dtx}
6 \end{document}
7 </driver>
```

## 39 The fonttext.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
8 ⟨*text⟩
9 \typeout{=== Don't modify this file, use a .cfg file instead ===^J}
```

### 39.1 Encodings

This file declares the standard encodings for text and math fonts. All others should be declared in packages or in the documents directly.

For every text encoding there are normally a number of encoding specific commands, e.g. accents, special characters, etc. (The definition for such a command might have to change when the encoding is changed, because the character is in a different position, or not available at all, or the accent is produced in a different way.) This is handled by a general mechanism which is described in `ltoutenc.dtx`.

By convention, text encoding specific declarations, including the declaration `\DeclareFontEncoding`, are kept in separate file of the form `⟨enc⟩enc.def`, e.g. `ot1enc.def`. This allows other applications to make use of the declarations as well.

Similar to the default encoding, the loading of the encoding files for the two major text encodings shouldn't be changed. In particular, the `inputenc` package depends on this.

```
10 \input {omlenc.def}
11 \input {t1enc.def}
12 \input {ot1enc.def} % <- should come after T1 for speed
13 \input {omsenc.def}
```

We then set set the default text font encoding. This will hopefully change some day to `T1`. This setting should *not* be changed to produce a portable format.

```
14 \fontencoding{OT1}
```

If different encodings for text fonts are in use one could put the common setup into `\DeclareFontEncodingDefaults`. There is now a better mechanism so using this interface is discouraged!

```
15 \DeclareFontEncodingDefaults{}{}
```

Then we define the default substitution for every encoding. This release of  $\text{\LaTeX} 2_{\epsilon}$  assumes that the ec fonts are available. It is possible to change this to point to some other font family (e.g., Times with the appropriate encoding if it is available) without making documents non-portable. However, in such a case documents will produce different page breaks at other sites. The substitution defaults can all be changed without losing portability as long as there are font shape definitions for the selected substitutions.

```
16 \DeclareFontSubstitution{T1}{cmr}{m}{n}
17 \DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration,  $\text{\LaTeX} 2_{\epsilon}$  will try to verify that the given substitution information makes sense, i.e. that it is impossible to go into an endless loop if font substitution happens. This is done at the moment the `\begin{document}` is encountered.  $\text{\LaTeX} 2_{\epsilon}$  will then check that for every encoding the substitution defaults form a valid font shape group, which means that it will check if there is a `\DeclareFontShape` declaration for this combination. We will therefore load the

corresponding `.fd` files now. If we don't do this they would be loaded at verification time (i.e. at `\begin{document}`) which would delay processing unnecessarily.

**Warning:** Please note that this means that you have to regenerate the format whenever you change any of these `.fd` files since  $\LaTeX 2_{\epsilon}$  will not read `.fd` files if it already knows about the encoding/family combination.

The `\nfss@catcodes` ensures that white space is ignored in any definitions made in the `fd` files.

```
18 \begingroup
19 \nfss@catcodes
20 \input {t1cmr.fd}
21 \input {ot1cmr.fd}
22 \endgroup
```

We also load some other font definition files which are normally needed in a document. This is only done for processing speed and you can comment the next two lines out to save some memory. If necessary these files are then loaded when your document is processed. (Loading `.fd` files is a less drastic step compared to preloading fonts because the number of fonts is limited 255 at (nearly) every  $\TeX$  installation, while the amount of main memory is not a limiting factor at most installations.)

```
23 \begingroup
24 \nfss@catcodes
25 \input {ot1cmss.fd}
26 \input {ot1cmtt.fd}
27 \endgroup
```

Even with all the precautions it is still possible that NFSS will run into problems, for example, when a `.fd` file contains corrupted data. To guard against such cases NFSS has a very low-level fallback font that is installed with the following line.

```
28 \DeclareErrorFont{OT1}{cmr}{m}{n}{10}
```

This means, “if everything else fails use Computer Modern Roman normal shape at 10pt in the old text encoding”. You can change the font used but the encoding should be the same as the one specified with `\fontencoding` above.

## 39.2 Defaults

To allow the use of `\rmfamily`, `\sffamily`, etc. in documents even if non-standard families are used we provide nine macros which hold the name of the corresponding families, series, and so on. This makes it easy to use other font families (like Times Roman, etc.). One simply has to redefine these defaults.

All these hooks have to be defined in this file but you can change their meaning (except for `\encodingdefault`) without making documents non-portable.

```
\rmdefault The following three definitions set up the meaning for \rmfamily, \sffamily, and
\sffdefault \ttfamily.
\ttdefault 29 \newcommand\rmdefault{cmr}
30 \newcommand\sffdefault{cmss}
31 \newcommand\ttdefault{cmtt}
```

```

\bfdefault Series changing commands are influenced by the following hooks.
\mddefault 32 \newcommand\bfdefault{bx}
 33 \newcommand\mddefault{m}

\itdefault Shape changing commands use the following hooks.
\sldefault 34 \newcommand\itdefault{it}
\scdefault 35 \newcommand\sldefault{sl}
\updefault 36 \newcommand\scdefault{sc}
 37 \newcommand\updefault{n}

\encodingdefault Finally we have the hooks that describe the behaviour of the \normalfont com-
\familydefault mand. To stay portable, the definition of \encodingdefault should not be
\seriesdefault changed and should match the setting above for \fontencoding. All other values
\shapedefault can be set according to your taste.
 38 \newcommand\encodingdefault{OT1}
 39 \newcommand\familydefault{\rmdefault}
 40 \newcommand\seriesdefault{\mddefault}
 41 \newcommand\shapedefault{\updefault}

 This finishes the low-level setup in fonttext.ltx.
 42 </text>

```

## 40 The fontmath.ltx file

The identification is done earlier on with a `\ProvidesFile` declaration.

```

43 (*math)
44 \typeout{=== Don't modify this file, use a .cfg file instead ===^^J}

```

### 40.1 The font encodings used

```

45 \DeclareFontEncoding{OML}{-}{-}
46 \DeclareFontEncoding{OMS}{-}{-}
47 \DeclareFontEncoding{OMX}{-}{-}

```

Finally a declaration for U encoding which serves for all fonts that do not fit standard encodings. For math this sets up `\noaccents@` providing for AMS- $\LaTeX$ . This macro is used therein to handle accented characters if they are not supported by the font. In other words, if fonts with U encoding are used in math, all accents (like from `\breve`) are obtained from some other font that has them.

```

48 \DeclareFontEncoding{U}{-}{\noaccents@}

```

The encodings for math are next:

```

49 \DeclareFontSubstitution{OML}{cmm}{m}{it}
50 \DeclareFontSubstitution{OMS}{cmsy}{m}{n}
51 \DeclareFontSubstitution{OMX}{cmex}{m}{n}
52 \DeclareFontSubstitution{U}{cmr}{m}{n}

53 \begingroup
54 \nfss@catcodes
55 \input {omlcmm.fd}
56 \input {omscmsy.fd}
57 \input {omxcmex.fd}
58 \input {ucmr.fd}

```

59 \endgroup

### 40.1.1 Symbolfont and Alphabet declarations

We now define the basic symbol fonts used by L<sup>A</sup>T<sub>E</sub>X. These four symbol fonts must be defined by this file.

It is possible to make the symbol fonts point to other external fonts without losing the ability to process documents written at other sites, as long as one defines the same symbol font names with the same encodings, e.g. `operators` with `OT1` etc. If other encodings are used documents become non-portable. Such a change should therefore be done in a package file.

```
60 \DeclareSymbolFont{operators} {OT1}{cmr} {m}{n}
61 \DeclareSymbolFont{letters} {OML}{cmm} {m}{it}
62 \DeclareSymbolFont{symbols} {OMS}{cmsy}{m}{n}
63 \DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}

64 \SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}
65 \SetSymbolFont{letters} {bold}{OML}{cmm} {b}{it}
66 \SetSymbolFont{symbols} {bold}{OMS}{cmsy}{b}{n}
```

Below are the seven math alphabets which are defined by NFSS. Again they must be defined by this file. However, as before you can change the fonts used without losing portability, but you should be careful when changing the encoding since that may make documents come out wrong.

```
67 \DeclareSymbolFontAlphabet{\mathrm} {operators}
68 \DeclareSymbolFontAlphabet{\mathnormal}{letters}
69 \DeclareSymbolFontAlphabet{\mathcal} {symbols}
70 \DeclareMathAlphabet {\mathbf} {OT1}{cmr}{bx}{n}
71 \DeclareMathAlphabet {\mathsf} {OT1}{cmss}{m}{n}
72 \DeclareMathAlphabet {\mathit} {OT1}{cmr}{m}{it}
73 \DeclareMathAlphabet {\mathtt} {OT1}{cmtt}{m}{n}
```

Given the currently available fonts we cannot bold-en `\mathbf` and `\mathtt` but in principle one could use ‘ultra bold’ or something. The alphabets defined via `\DeclareSymbolFontAlphabet` will change automatically in a new math version if the corresponding symbol font changes.

```
74 \SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
75 \SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}
```

## 40.2 Math font sizes

The declarations below declare the text, script and scriptscript size to be used for each text font size.

All occurrences of sizes longer than a single character are replaced with the macro name that holds them, saving a number of tokens (but losing a bit of speed, so this may not stay this way).

```
76 \DeclareMathSizes{5}{5}{5}{5}
77 \DeclareMathSizes{6}{6}{5}{5}
78 \DeclareMathSizes{7}{7}{5}{5}
79 \DeclareMathSizes{8}{8}{6}{5}
80 \DeclareMathSizes{9}{9}{6}{5}
81 \DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
82 \DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
83 \DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
```

```

84 \DeclareMathSizes{\xivpt}{\xivpt}{\xpt}{7}
85 \DeclareMathSizes{\xviipt}{\xviipt}{\xiipt}{\xpt}
86 \DeclareMathSizes{\xxxpt}{\xxxpt}{\xivpt}{\xiipt}
87 \DeclareMathSizes{\xxvpt}{\xxvpt}{\xxxpt}{\xviipt}

```

### 40.3 The math symbol assignments

We start by setting up math codes for most of the characters typed in directly from the keyboard. Most of them are normally already setup up in the same way by `IniTEX`. However, we repeat them here to have a complete setup which can be exchanged with another if desired.

#### 40.3.1 The letters

```

88 \DeclareMathSymbol{a}{\mathalpha}{letters}{'a}
89 \DeclareMathSymbol{b}{\mathalpha}{letters}{'b}
90 \DeclareMathSymbol{c}{\mathalpha}{letters}{'c}
91 \DeclareMathSymbol{d}{\mathalpha}{letters}{'d}
92 \DeclareMathSymbol{e}{\mathalpha}{letters}{'e}
93 \DeclareMathSymbol{f}{\mathalpha}{letters}{'f}
94 \DeclareMathSymbol{g}{\mathalpha}{letters}{'g}
95 \DeclareMathSymbol{h}{\mathalpha}{letters}{'h}
96 \DeclareMathSymbol{i}{\mathalpha}{letters}{'i}
97 \DeclareMathSymbol{j}{\mathalpha}{letters}{'j}
98 \DeclareMathSymbol{k}{\mathalpha}{letters}{'k}
99 \DeclareMathSymbol{l}{\mathalpha}{letters}{'l}
100 \DeclareMathSymbol{m}{\mathalpha}{letters}{'m}
101 \DeclareMathSymbol{n}{\mathalpha}{letters}{'n}
102 \DeclareMathSymbol{o}{\mathalpha}{letters}{'o}
103 \DeclareMathSymbol{p}{\mathalpha}{letters}{'p}
104 \DeclareMathSymbol{q}{\mathalpha}{letters}{'q}
105 \DeclareMathSymbol{r}{\mathalpha}{letters}{'r}
106 \DeclareMathSymbol{s}{\mathalpha}{letters}{'s}
107 \DeclareMathSymbol{t}{\mathalpha}{letters}{'t}
108 \DeclareMathSymbol{u}{\mathalpha}{letters}{'u}
109 \DeclareMathSymbol{v}{\mathalpha}{letters}{'v}
110 \DeclareMathSymbol{w}{\mathalpha}{letters}{'w}
111 \DeclareMathSymbol{x}{\mathalpha}{letters}{'x}
112 \DeclareMathSymbol{y}{\mathalpha}{letters}{'y}
113 \DeclareMathSymbol{z}{\mathalpha}{letters}{'z}

114 \DeclareMathSymbol{A}{\mathalpha}{letters}{'A}
115 \DeclareMathSymbol{B}{\mathalpha}{letters}{'B}
116 \DeclareMathSymbol{C}{\mathalpha}{letters}{'C}
117 \DeclareMathSymbol{D}{\mathalpha}{letters}{'D}
118 \DeclareMathSymbol{E}{\mathalpha}{letters}{'E}
119 \DeclareMathSymbol{F}{\mathalpha}{letters}{'F}
120 \DeclareMathSymbol{G}{\mathalpha}{letters}{'G}
121 \DeclareMathSymbol{H}{\mathalpha}{letters}{'H}
122 \DeclareMathSymbol{I}{\mathalpha}{letters}{'I}
123 \DeclareMathSymbol{J}{\mathalpha}{letters}{'J}
124 \DeclareMathSymbol{K}{\mathalpha}{letters}{'K}
125 \DeclareMathSymbol{L}{\mathalpha}{letters}{'L}
126 \DeclareMathSymbol{M}{\mathalpha}{letters}{'M}

```

```

127 \DeclareMathSymbol{N}{\mathalpha}{letters}{'N}
128 \DeclareMathSymbol{O}{\mathalpha}{letters}{'O}
129 \DeclareMathSymbol{P}{\mathalpha}{letters}{'P}
130 \DeclareMathSymbol{Q}{\mathalpha}{letters}{'Q}
131 \DeclareMathSymbol{R}{\mathalpha}{letters}{'R}
132 \DeclareMathSymbol{S}{\mathalpha}{letters}{'S}
133 \DeclareMathSymbol{T}{\mathalpha}{letters}{'T}
134 \DeclareMathSymbol{U}{\mathalpha}{letters}{'U}
135 \DeclareMathSymbol{V}{\mathalpha}{letters}{'V}
136 \DeclareMathSymbol{W}{\mathalpha}{letters}{'W}
137 \DeclareMathSymbol{X}{\mathalpha}{letters}{'X}
138 \DeclareMathSymbol{Y}{\mathalpha}{letters}{'Y}
139 \DeclareMathSymbol{Z}{\mathalpha}{letters}{'Z}

```

### 40.3.2 The digits

```

140 \DeclareMathSymbol{0}{\mathalpha}{operators}{'0}
141 \DeclareMathSymbol{1}{\mathalpha}{operators}{'1}
142 \DeclareMathSymbol{2}{\mathalpha}{operators}{'2}
143 \DeclareMathSymbol{3}{\mathalpha}{operators}{'3}
144 \DeclareMathSymbol{4}{\mathalpha}{operators}{'4}
145 \DeclareMathSymbol{5}{\mathalpha}{operators}{'5}
146 \DeclareMathSymbol{6}{\mathalpha}{operators}{'6}
147 \DeclareMathSymbol{7}{\mathalpha}{operators}{'7}
148 \DeclareMathSymbol{8}{\mathalpha}{operators}{'8}
149 \DeclareMathSymbol{9}{\mathalpha}{operators}{'9}

```

### 40.3.3 Punctuation, brace, etc. keys

```

150 \DeclareMathSymbol{!}{\mathclose}{operators}{"21}
151 \DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
152 \DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
153 \DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
154 \DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
155 \DeclareMathSymbol{.}{\mathord}{letters}{"3A}
156 \DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
157 \DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
158 \DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
159 \DeclareMathSymbol{?}{\mathclose}{operators}{"3F}

```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```

160 %\DeclareMathSymbol{(}{\mathopen}{operators}{"28}
161 %\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
162 %\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
163 %\DeclareMathSymbol{[}{\mathopen}{operators}{"5B}
164 %\DeclareMathSymbol{]}{\mathclose}{operators}{"5D}
165 %\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
166 %\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
167 %\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}

```

Should all of the following being activated by default? Probably not.

```

168 %\DeclareMathSymbol{\}{\mathopen}{symbols}{"66}
169 %\DeclareMathSymbol{\'}{\mathclose}{symbols}{"67}
170 %\DeclareMathSymbol{\}\'}{\mathord}{symbols}{"6E} % \backslash
171 \mathcode\` = "8000 % \space
172 \mathcode\`' = "8000 % ^\prime

```

```
173 \mathcode'_="8000 % _
```

#### 40.3.4 Delimitercodes for characters

[to be completed]

Finally, Ini $\TeX$  sets all `\delcode` values to -1, except `\delcode' .=0`

```
174 \DeclareMathDelimiter{(\mathopen}{operators}{"28}{largesymbols}{"00}
175 \DeclareMathDelimiter{)\mathclose}{operators}{"29}{largesymbols}{"01}
176 \DeclareMathDelimiter{[\mathopen}{operators}{"5B}{largesymbols}{"02}
177 \DeclareMathDelimiter{]\mathclose}{operators}{"5D}{largesymbols}{"03}
```

The next two are considered to be relations when not used in the context of a delimiter! And worse, they do even represent different glyphs when being used as delimiter and not as delimiter. This is a user level syntax inherited from plain  $\TeX$ . Therefore we explicitly redefine the math symbol definitions for these symbols afterwards.

```
178 \DeclareMathDelimiter{<\mathopen}{symbols}{"68}{largesymbols}{"0A}
179 \DeclareMathDelimiter{>\mathclose}{symbols}{"69}{largesymbols}{"0B}
180 \DeclareMathSymbol{<\mathrel}{letters}{"3C}
181 \DeclareMathSymbol{>\mathrel}{letters}{"3E}
```

And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```
182 \DeclareMathDelimiter{/}\mathord}{operators}{"2F}{largesymbols}{"0E}
183 \DeclareMathSymbol{/}\mathord}{letters}{"3D}
184 \DeclareMathDelimiter{|\mathord}{symbols}{"6A}{largesymbols}{"0C}
185 \expandafter\DeclareMathDelimiter\@backslashchar
186 \mathord}{symbols}{"6E}{largesymbols}{"0F}
```

N.B. { and } should NOT get delcodes; otherwise parameter grouping fails!

## 40.4 Symbols accessed via control sequences

### 40.4.1 Greek letters

```
187 \DeclareMathSymbol{\alpha}\mathord}{letters}{"0B}
188 \DeclareMathSymbol{\beta}\mathord}{letters}{"0C}
189 \DeclareMathSymbol{\gamma}\mathord}{letters}{"0D}
190 \DeclareMathSymbol{\delta}\mathord}{letters}{"0E}
191 \DeclareMathSymbol{\epsilon}\mathord}{letters}{"0F}
192 \DeclareMathSymbol{\zeta}\mathord}{letters}{"10}
193 \DeclareMathSymbol{\eta}\mathord}{letters}{"11}
194 \DeclareMathSymbol{\theta}\mathord}{letters}{"12}
195 \DeclareMathSymbol{\iota}\mathord}{letters}{"13}
196 \DeclareMathSymbol{\kappa}\mathord}{letters}{"14}
197 \DeclareMathSymbol{\lambda}\mathord}{letters}{"15}
198 \DeclareMathSymbol{\mu}\mathord}{letters}{"16}
199 \DeclareMathSymbol{\nu}\mathord}{letters}{"17}
200 \DeclareMathSymbol{\xi}\mathord}{letters}{"18}
201 \DeclareMathSymbol{\pi}\mathord}{letters}{"19}
202 \DeclareMathSymbol{\rho}\mathord}{letters}{"1A}
203 \DeclareMathSymbol{\sigma}\mathord}{letters}{"1B}
204 \DeclareMathSymbol{\tau}\mathord}{letters}{"1C}
205 \DeclareMathSymbol{\upsilon}\mathord}{letters}{"1D}
206 \DeclareMathSymbol{\phi}\mathord}{letters}{"1E}
```



```

207 \DeclareMathSymbol{\chi}{\mathord}{letters}{"1F}
208 \DeclareMathSymbol{\psi}{\mathord}{letters}{"20}
209 \DeclareMathSymbol{\omega}{\mathord}{letters}{"21}
210 \DeclareMathSymbol{\varepsilon}{\mathord}{letters}{"22}
211 \DeclareMathSymbol{\vartheta}{\mathord}{letters}{"23}
212 \DeclareMathSymbol{\varpi}{\mathord}{letters}{"24}
213 \DeclareMathSymbol{\varrho}{\mathord}{letters}{"25}
214 \DeclareMathSymbol{\varsigma}{\mathord}{letters}{"26}
215 \DeclareMathSymbol{\varphi}{\mathord}{letters}{"27}
216 \DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{"00}
217 \DeclareMathSymbol{\Delta}{\mathalpha}{operators}{"01}
218 \DeclareMathSymbol{\Theta}{\mathalpha}{operators}{"02}
219 \DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{"03}
220 \DeclareMathSymbol{\Xi}{\mathalpha}{operators}{"04}
221 \DeclareMathSymbol{\Pi}{\mathalpha}{operators}{"05}
222 \DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{"06}
223 \DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{"07}
224 \DeclareMathSymbol{\Phi}{\mathalpha}{operators}{"08}
225 \DeclareMathSymbol{\Psi}{\mathalpha}{operators}{"09}
226 \DeclareMathSymbol{\Omega}{\mathalpha}{operators}{"0A}

```

#### 40.4.2 Ordinary symbols

```

227 \DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
228 \def\hbar{{\mathchar'26\mkern-9mu\h}}
229 \DeclareMathSymbol{\imath}{\mathord}{letters}{"7B}
230 \DeclareMathSymbol{\jmath}{\mathord}{letters}{"7C}
231 \DeclareMathSymbol{\ell}{\mathord}{letters}{"60}
232 \DeclareMathSymbol{\wp}{\mathord}{letters}{"7D}
233 \DeclareMathSymbol{\Re}{\mathord}{symbols}{"3C}
234 \DeclareMathSymbol{\Im}{\mathord}{symbols}{"3D}
235 \DeclareMathSymbol{\partial}{\mathord}{letters}{"40}
236 \DeclareMathSymbol{\infty}{\mathord}{symbols}{"31}
237 \DeclareMathSymbol{\prime}{\mathord}{symbols}{"30}
238 \DeclareMathSymbol{\emptyset}{\mathord}{symbols}{"3B}
239 \DeclareMathSymbol{\nabla}{\mathord}{symbols}{"72}
240 \def\surd{{\mathchar"1270}}
241 \DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
242 \DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
243 \def\angle{{\vbox{\ialign{${\m@th\scriptstyle##}\cr
244 \not\mathrel{\mkern14mu}\cr
245 \noalign{\nointerlineskip}
246 \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\cr}}}
247 \DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
248 \DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
249 \DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
250 \DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}
251 \let\not=\neg
252 \DeclareMathSymbol{\flat}{\mathord}{letters}{"5B}
253 \DeclareMathSymbol{\natural}{\mathord}{letters}{"5C}
254 \DeclareMathSymbol{\sharp}{\mathord}{letters}{"5D}
255 \DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{"7C}
256 \DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{"7D}
257 \DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{"7E}
258 \DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{"7F}

```

### 40.4.3 Large Operators

```
259 \DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{"60}
260 \DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{"57}
261 \DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{"56}
262 \DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{"55}
263 \DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{"54}
264 \DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{"53}
265 \DeclareMathSymbol{\intop}{\mathop}{largesymbols}{"52}
266 \def\int{\intop\nolimits}
267 \DeclareMathSymbol{\prod}{\mathop}{largesymbols}{"51}
268 \DeclareMathSymbol{\sum}{\mathop}{largesymbols}{"50}
269 \DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{"4E}
270 \DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{"4C}
271 \DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{"4A}
272 \DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{"48}
273 \def\oint{\ointop\nolimits}
274 \DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{"46}
275 \DeclareMathSymbol{\smallint}{\mathop}{symbols}{"73}
```

### 40.4.4 Binary symbols

```
276 \DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{"2F}
277 \DeclareMathSymbol{\triangleright}{\mathbin}{letters}{"2E}
278 \DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{"34}
279 \DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{"35}
280 \let \varbigtriangledown \bigtriangledown
281 \let \varbigtriangleup \bigtriangleup
```

These last two synonyms are needed because the `stamryrd` package redefines them as Operators.

```
282 \DeclareMathSymbol{\wedge}{\mathbin}{symbols}{"5E}
283 \let\land=\wedge
284 \DeclareMathSymbol{\vee}{\mathbin}{symbols}{"5F}
285 \let\lor=\vee
286 \DeclareMathSymbol{\cap}{\mathbin}{symbols}{"5C}
287 \DeclareMathSymbol{\cup}{\mathbin}{symbols}{"5B}
288 \DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{"7A}
289 \DeclareMathSymbol{\dagger}{\mathbin}{symbols}{"79}
290 \DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{"75}
291 \DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{"74}
292 \DeclareMathSymbol{\uplus}{\mathbin}{symbols}{"5D}
293 \DeclareMathSymbol{\amalg}{\mathbin}{symbols}{"71}
294 \DeclareMathSymbol{\diamond}{\mathbin}{symbols}{"05}
295 \DeclareMathSymbol{\bullet}{\mathbin}{symbols}{"0F}
296 \DeclareMathSymbol{\wr}{\mathbin}{symbols}{"6F}
297 \DeclareMathSymbol{\div}{\mathbin}{symbols}{"04}
298 \DeclareMathSymbol{\odot}{\mathbin}{symbols}{"0C}
299 \DeclareMathSymbol{\oslash}{\mathbin}{symbols}{"0B}
300 \DeclareMathSymbol{\otimes}{\mathbin}{symbols}{"0A}
301 \DeclareMathSymbol{\ominus}{\mathbin}{symbols}{"09}
302 \DeclareMathSymbol{\oplus}{\mathbin}{symbols}{"08}
303 \DeclareMathSymbol{\mp}{\mathbin}{symbols}{"07}
304 \DeclareMathSymbol{\pm}{\mathbin}{symbols}{"06}
305 \DeclareMathSymbol{\circ}{\mathbin}{symbols}{"0E}
306 \DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{"0D}
```

```

307 \DeclareMathSymbol{\setminus}{\mathbin}{symbols}{"6E}
308 \DeclareMathSymbol{\cdot}{\mathbin}{symbols}{"01}
309 \DeclareMathSymbol{\ast}{\mathbin}{symbols}{"03}
310 \DeclareMathSymbol{\times}{\mathbin}{symbols}{"02}
311 \DeclareMathSymbol{\star}{\mathbin}{letters}{"3F}

```

#### 40.4.5 Relations

```

312 \DeclareMathSymbol{\propto}{\mathrel}{symbols}{"2F}
313 \DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{"76}
314 \DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{"77}
315 \DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
316 \DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
317 \DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
318 \DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
319 \DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
320 \DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
321 \DeclareMathSymbol{\nrightarrow}{\mathrel}{symbols}{"2D}
322 \DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
323 \DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
324 \DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
325 \DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
326 \def\neq{\not=} \let\ne=\neq
327 \DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
328 \let\le=\leq
329 \DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}
330 \let\ge=\geq
331 \DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
332 \DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
333 \DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
334 \DeclareMathSymbol{\succeeds}{\mathrel}{symbols}{"17}
335 \DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
336 \DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
337 \DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
338 \DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
339 \DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
340 \DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
341 \DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}
342 \let\owns=\ni
343 \DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
344 \DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
345 \DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
346 \DeclareMathSymbol{\leftrightharpoonup}{\mathrel}{symbols}{"24}
347 \DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
348 \let\gets=\leftarrow
349 \DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}
350 \let\to=\rightarrow
351 \DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
352 \def\mapsto{\mapstochar\rightarrow}
353 \DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
354 \DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
355 \DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
356 \DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
357 \DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
358 \DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}

```

```

359 \DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
360 \DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
361 \DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
362 \DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
363 \DeclareMathSymbol{\rightharpoondown}{\mathrel}{letters}{"2B}

```

Here cometh much profligate robustification of math constructs. Warning: some of these commands may become non-robust if an AMS package is loaded.

Further potential problems: some math font packages may make unfortunate assumptions about some of these definitions that are not true of the robust versions we need.

```

364 \DeclareRobustCommand
365 \cong{\mathrel{\mathpalette@vereq\sim}} % congruence sign
366 \def\@vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
367 \ialign{\$m@th#1\hfil#\hfil$\crcr#2\crcr=\crcr}}}
368 \DeclareRobustCommand
369 \notin{\mathrel{\m@th\mathpalette@cncel\in}}
370 \def\cncel#1#2{\m@th\oalign{\$hfil#1\mkern1mu/hfil$\crcr#1#2$}}
371 \DeclareRobustCommand
372 \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
373 \def\rlh@#1{\vcenter{\m@th\hbox{\oalign{\raise2pt
374 \hbox{#1\rightharpoonup}\crcr
375 #1\leftharpoondown}}}}
376 \DeclareRobustCommand
377 \doteq{\buildrel\textstyle.\over=}

```

#### 40.4.6 Arrows

```

378 \DeclareRobustCommand
379 \joinrel{\mathrel{\mkern-3mu}}
380 \DeclareRobustCommand
381 \relbar{\mathrel{\smash-}} % \smash, because -
382 % has the same height as +

```

In contrast to `plain.tex` `\Relbar` got braces around the equal sign to guard against it being “math active” expanding to `\futurelet...`. This might be the case when packages are implementing shorthands for math, e.g. `=>` meaning `\Rightarrow` etc. It would actually be better not to use `=` in such definitions but instead define something like `\mathequalsign` and use this. However we can’t do this now as it would break other math layouts where characters are in different places (since those wouldn’t know about the need for a new command name).

```

383 \DeclareRobustCommand
384 \Relbar{\mathrel{=}}
385 \DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
386 \def\hookrightarrow{\lhook\joinrel\rightarrow}
387 \DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
388 \def\hookleftarrow{\leftarrow\joinrel\rhook}
389 \DeclareRobustCommand
390 \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
391 \DeclareRobustCommand
392 \models{\mathrel{||}\joinrel\Relbar}
393 \DeclareRobustCommand
394 \Longrightarrow{\Relbar\joinrel\rightarrow}

```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make them robust.

```

395 \DeclareRobustCommand\longrightarrow
396 {\relbar\joinrel\rightarrow}
397 \DeclareRobustCommand\longleftarrow
398 {\leftarrow\joinrel\relbar}
399 \DeclareRobustCommand
400 \Longleftarrow{\Leftarrow\joinrel\Relbar}
401 \DeclareRobustCommand
402 \longmapsto{\mapstochar\longrightarrow}
403 \DeclareRobustCommand
404 \longlefttrightarrow{\leftarrow\joinrel\rightarrow}
405 \DeclareRobustCommand
406 \Longlefttrightarrow{\Leftarrow\joinrel\Rightarrow}
407 \DeclareRobustCommand
408 \iff{\;\Longlefttrightarrow\;}

```

#### 40.4.7 Punctuation symbols

```

409 \DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
410 \DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
411 \DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}

```

This is commented out, since `\ldots` is now defined in `ltoutenc.dtx`.

```

412 %\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
413 %\DeclareRobustCommand\ldots
414 % {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots$,}$\fi}
415 \DeclareRobustCommand
416 \cdots{\mathinner{\cdotp\cdotp\cdotp}}
417 \DeclareRobustCommand
418 \vdots{\vbox{\baselineskip4\p@ \lineskiplimit\z@
419 \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
420 \DeclareRobustCommand
421 \ddots{\mathinner{\mkern1mu\raise7\p@
422 \vbox{\kern7\p@\hbox{.}}\mkern2mu
423 \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}

```

#### 40.4.8 Math accents

```

424 \DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
425 \DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}
426 \DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
427 \DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
428 \DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
429 \DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
430 \DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
431 \DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
432 \DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
433 \DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
434 \DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
435 \DeclareMathAccent{\widehat}{\mathord}{largesymbols}{"62}

```

For some reason plain T<sub>E</sub>X never bothered to provide a ring accent in math (although it is available in the fonts), but since we got a request for it here we go:

```

436 \DeclareMathAccent{\mathring}{\mathalpha}{operators}{"17}

```

#### 40.4.9 Radicals

437 \DeclareMathRadical{\sqrtsgn}{symbols}{"70}{largesymbols}{"70}

#### 40.4.10 Over and under something, etc

```

438 \def\overrightarrow#1{\vbox{\m@th\ialign{##\crrc
439 \rightarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}
440 $\hfil\displaystyle{#1}\hfil$\crrc}}}
441 \def\overleftarrow#1{\vbox{\m@th\ialign{##\crrc
442 \leftarrowfill\crrc\noalign{\kern-\p@\nointerlineskip}%
443 $\hfil\displaystyle{#1}\hfil$\crrc}}}
444 \def\overbrace#1{\mathop{\vbox{\m@th\ialign{##\crrc\noalign{\kern3\p@}%
445 \downbracefill\crrc\noalign{\kern3\p@\nointerlineskip}%
446 $\hfil\displaystyle{#1}\hfil$\crrc}}}\limits}
447 \def\underbrace#1{\mathop{\vtop{\m@th\ialign{##\crrc
448 $\hfil\displaystyle{#1}\hfil$\crrc
449 \noalign{\kern3\p@\nointerlineskip}%
450 \upbracefill\crrc\noalign{\kern3\p@}}}\limits}

```

(quite a waste of tokens, IMHO — Frank)

```

451 \def\skew#1#2#3{\muskip\z@#1mu\divide\muskip\z@tw@ \mkern\muskip\z@
452 #2{\mkern-\muskip\z@#3}\mkern\muskip\z@}\mkern-\muskip\z@{}}
453 \def\rightarrowfill{\m@th\smash-\mkern-7mu%
454 \cleaders\hbox{\m@th\smash-\mkern-2mu$}\hfill
455 \mkern-7mu\mathord\rightarrow$}
456 \def\leftarrowfill{\m@th\mathord\leftarrow\mkern-7mu%
457 \cleaders\hbox{\m@th\smash-\mkern-2mu$}\hfill
458 \mkern-7mu\smash-$}
459 \DeclareMathSymbol{\bracedl}{\mathord}{largesymbols}{"7A}
460 \DeclareMathSymbol{\bracerd}{\mathord}{largesymbols}{"7B}
461 \DeclareMathSymbol{\bracelu}{\mathord}{largesymbols}{"7C}
462 \DeclareMathSymbol{\braceru}{\mathord}{largesymbols}{"7D}
463 \def\downbracefill{\m@th \setbox\z@\hbox{\m@th\bracedl}$}
464 \bracedl\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru
465 \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd$}
466 \def\upbracefill{\m@th \setbox\z@\hbox{\m@th\bracedl}$}
467 \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
468 \bracedl\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}

```

#### 40.4.11 Delimiters

```

469 \DeclareMathDelimiter{\lmoustache} % top from (, bottom from)
470 {\mathopen}{largesymbols}{"7A}{largesymbols}{"40}
471 \DeclareMathDelimiter{\rmoustache} % top from), bottom from (
472 {\mathclose}{largesymbols}{"7B}{largesymbols}{"41}
473 \DeclareMathDelimiter{\arrowvert} % arrow without arrowheads
474 {\mathord}{symbols}{"6A}{largesymbols}{"3C}
475 \DeclareMathDelimiter{\Arrowvert} % double arrow without arrowheads
476 {\mathord}{symbols}{"6B}{largesymbols}{"3D}
477 \DeclareMathDelimiter{\Vert}
478 {\mathord}{symbols}{"6B}{largesymbols}{"0D}
479 \let\|= \Vert
480 \DeclareMathDelimiter{\vert}
481 {\mathord}{symbols}{"6A}{largesymbols}{"0C}
482 \DeclareMathDelimiter{\uparrow}
483 {\mathrel}{symbols}{"22}{largesymbols}{"78}
484 \DeclareMathDelimiter{\downarrow}

```

```

485 {\mathrel}{symbols}{"23}{largesymbols}{"79}
486 \DeclareMathDelimiter{\updownarrow}
487 {\mathrel}{symbols}{"6C}{largesymbols}{"3F}
488 \DeclareMathDelimiter{\Uparrow}
489 {\mathrel}{symbols}{"2A}{largesymbols}{"7E}
490 \DeclareMathDelimiter{\Downarrow}
491 {\mathrel}{symbols}{"2B}{largesymbols}{"7F}
492 \DeclareMathDelimiter{\Updownarrow}
493 {\mathrel}{symbols}{"6D}{largesymbols}{"77}
494 \DeclareMathDelimiter{\backslash} % for double coset G\backslash H
495 {\mathord}{symbols}{"6E}{largesymbols}{"0F}
496 \DeclareMathDelimiter{\rangle}
497 {\mathclose}{symbols}{"69}{largesymbols}{"0B}
498 \DeclareMathDelimiter{\langle}
499 {\mathopen}{symbols}{"68}{largesymbols}{"0A}
500 \DeclareMathDelimiter{\rbrace}
501 {\mathclose}{symbols}{"67}{largesymbols}{"09}
502 \DeclareMathDelimiter{\lbrace}
503 {\mathopen}{symbols}{"66}{largesymbols}{"08}
504 \DeclareMathDelimiter{\rceil}
505 {\mathclose}{symbols}{"65}{largesymbols}{"07}
506 \DeclareMathDelimiter{\lceil}
507 {\mathopen}{symbols}{"64}{largesymbols}{"06}
508 \DeclareMathDelimiter{\rfloor}
509 {\mathclose}{symbols}{"63}{largesymbols}{"05}
510 \DeclareMathDelimiter{\lfloor}
511 {\mathopen}{symbols}{"62}{largesymbols}{"04}

```

`\lgroup` There are three plain T<sub>E</sub>X delimiters which are not fully supported by NFSS, since they partly point into a bold cmr font. Allocating a full symbol font, just `\rgroup` to have three delimiters seems a bit too much given the limited space available. `\bracevert` For this reason only the extensible sizes are supported. If this is not desired one can use, without losing portability, define `\mathbf` and `\mathtt` as font symbol alphabet (setting up `cmr/bx/n` and `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.)

```

512 \DeclareMathDelimiter{\lgroup} % extensible (with sharper tips
513 {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
514 \DeclareMathDelimiter{\rgroup} % extensible) with sharper tips
515 {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}
516 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
517 {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}

```

## 40.5 Math versions of text commands

The `\mathunderscore` here is really a text definition, so it has been put back into `ltoutenc.dtx` (by Chris, 30/04/97) and should be removed from here.

These symbols are the math versions of text commands such as `\P`, `\$`, etc.

```

\mathparagraph These math symbols are not in plain TEX.
\mathsection 518 \DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
\mathdollar 519 \DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
\mathsterling 520 \DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
\mathunderscore

```

```
521 \def\mathsterling{\mathit{\mathchar"7024}}
522 \def\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}
```

`\mathellipsis` This is plain T<sub>E</sub>X's `\ldots`.

```
523 \def\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%
```

## 40.6 Other special functions and parameters

### 40.6.1 Biggggg

```
524 \def\big#1{{\hbox{$\left#1\vbox to8.5\p@{\right.\n@space$}}}
525 \def\Big#1{{\hbox{$\left#1\vbox to11.5\p@{\right.\n@space$}}}
526 \def\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{\right.\n@space$}}}
527 \def\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{\right.\n@space$}}}
528 \def\n@space{\null\delimiterspace\z@ \m@th}
```

### 40.6.2 The log-like functions

`\operator@font` The `\operator@font` determines the symbol font used for log-like functions.

```
529 \def\operator@font{\mathgroup\symoperators}
```

### 40.6.3 Parameters

```
530 \thinmuskip=3mu
531 \medmuskip=4mu plus 2mu minus 4mu
532 \thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in `fontmath.ltx`.

```
533 </math>
```

## 41 Default cfg files

We provide default `cfg` files here to ensure that on installations that search large file trees we do not pick up some strange customisation files from somewhere.

```
534 (*cfgtext | cfgmath | cfgprel)
535 %%
536 %%
537 %%
538 %% Load the standard setup:
539 %%
540 <+cfgtext>\input{fonttext.ltx}
541 <+cfgmath>\input{fontmath.ltx}
542 <+cfgprel>\input{preload.ltx}
543 %%
544 %% Small changes could go here; see documentation in cfgguide.tex for
545 %% allowed modifications.
546 %%
547 %% In particular it is not allowed to misuse this configuration file
548 %% to modify internal LaTeX commands!
549 %%
550 %% If you use this file as the basis for configuration please change
551 %% the \ProvidesFile lines to clearly identify your modification, e.g.,
552 %%
553 <+cfgtext>%% \ProvidesFile{fonttext.cfg}[2001/06/01
554 <+cfgmath>%% \ProvidesFile{fonttext.cfg}[2001/06/01
```



```
555 (+cfgprel)%% \ProvidesFile{preload.cfg}[2001/06/01
556 %% Customised local font setup]
557 %%
558 %%
559 (/cfgtext | cfgmath | cfgprel)
```

## File u

# preload.dtx

## 42 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by  $\text{\LaTeX} 2_{\epsilon}$ ). 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.

## 43 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.

## 44 Module switches for the DOCSTRIP program

The DOCSTRIP will generate the above file from this source using the following module directives:

|         |                                                         |
|---------|---------------------------------------------------------|
| driver  | produce a documentation driver file                     |
| preload | produce a preload... file                               |
| cm      | for OT1 encoded Computer Modern                         |
| dc      | for T1 encoded Computer Modern                          |
| min     | produce minimal subset                                  |
| xpt     | produce 10pt preloads                                   |
| xipt    | produce 11pt preloads                                   |
| xiipt   | produce 12pt preloads                                   |
| ori     | produce preloads similar to old <code>lfonts.tex</code> |
| tex     | produce <code>preload.ltx</code>                        |

A typical DOCSTRIP command file would then have entries like:

```
\generateFile{preload.min}{t}{\from{preload.dtx}{preload,min}}
```

for generating preload files.

## 45 A driver for this document

The next bit of code contains the documentation driver file for  $\TeX$ , 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>
```

## 46 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\tenlcirc=lcircle10 \font\tenlcircw=lcirclew10\relax
```

The above fonts should not be touched but anything below this point here in the preload suggestions can be modified without any problems.

```
11 (-tex)%*****
```

```

12 (-tex)% Start any modification below this point **
13 (-tex)%*****
14 (-tex)
15 %%
16 %% Computer Modern Roman:
17 %%-----
18 (*ori)
19 \DeclarePreloadSizes{OT1}{cmr}{m}{n}
20 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
21 \DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
22 \DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
23 \DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
24 (/ori)
25 (+xpt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
26 (+xpt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
27 (+xipt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
28 (+xipt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
29 (+xiipt & cm) \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
30 (+xiipt & dc) \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
31 %%
32 %% Computer Modern Sans:
33 %%-----
34 (+ori) \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
35 %%
36 %% Computer Modern Typewriter:
37 %%-----
38 (+ori) \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
39 %%
40 %% Computer Modern Math:
41 %%-----
42 (*ori)
43 \DeclarePreloadSizes{OML}{cmm}{m}{it}
44 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
45 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}
46 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
47 (/ori)

```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```

48 (*xpt)
49 \DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
50 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
51 (/xpt)
52 (*xipt)
53 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
54 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
55 (/xipt)
56 (*xiipt)
57 \DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
58 \DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
59 (/xiipt)
60 %%
61 %% LaTeX symbol fonts:
62 %%-----

```

```
63 <*ori>
64 \DeclarePreloadSizes{U}{lasy}{m}{n}
65 {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
66 </ori>
67 </preload>
```

# File v

## 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.

## 47 Introduction

Font changes such as `\bfseries`, `\sffamily`, etc. are declarations; this means that their scope is delimited by the grouping structure, either by the next `\end` of some environment or by explicitly using a group, e.g., writing something like `{\bfseries...}` in the source. If you make the mistake of writing `\bfseries{...}` (thinking of `\bfseries` as a command with one argument) then the result is rather striking.

Font declarations are an artifact of the  $\text{T}_{\text{E}}\text{X}$  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`.<sup>7</sup> The default definition for this list is produced by the following.

```
\newcommand \nocorrlist {,.}
```

---

<sup>7</sup>Any package that changes the `\catcode` of a character inside `\nocorrlist` must then explicitly reset the list. Otherwise the changed character will no longer be recognized by the suppression algorithm.

| <i>Command</i>           | <i>Corresponds to</i>  | <i>Action</i>                            |
|--------------------------|------------------------|------------------------------------------|
| <code>\textrm{..}</code> | <code>\rmfamily</code> | Typeset argument in roman family         |
| <code>\textsf{..}</code> | <code>\sffamily</code> | Typeset argument in 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 <b>bold</b> series   |
| <code>\textup{..}</code> | <code>\upshape</code>  | Typeset argument in normal shape         |
| <code>\textit{..}</code> | <code>\itshape</code>  | Typeset argument in <i>italic</i> shape  |
| <code>\textsl{..}</code> | <code>\slshape</code>  | Typeset argument in <i>slanted</i> shape |
| <code>\textsc{..}</code> | <code>\scshape</code>  | Typeset argument in SMALL CAPS shape     |
| <code>\emph{..}</code>   | <code>\em</code>       | Typeset argument <i>emphasized</i>       |

Table 1: Font-change commands with arguments

The font change commands provided here all start with `\text..` to emphasize that they are for use in normal text and to be easily memorable. They automatically take care of any necessary italic correction on either side of the argument.

It is best to declare the most often used characters first, because this will make the processing slightly faster. For example,

```
\emph{When using the \NFSS{} high-level commands,
the \emph{proper} use of italic corrections is
automatically taken care of}. Only
\emph{sometimes} one has to help \LaTeX{} by
adding a \verb=\nocorr= command.
```

which results in:

*When using the NFSS high-level commands, the proper use of italic corrections is automatically taken care of. Only sometimes one has to help L<sup>A</sup>T<sub>E</sub>X by adding a \nocorr command.*

In contrast, the use of the declaration forms is often more appropriate when you define your own commands or environments.

```
\newenvironment{bfitemize}{\begin{itemize}\normalfont\bfseries}
{\end{itemize}}
\begin{bfitemize}
\item This environment produces boldface items.
\item It is defined in terms of \LaTeX's
\texttt{itemize} environment and NFSS
declarations.
\end{bfitemize}
```

This gives:

- **This environment produces boldface items.**

- **It is defined in terms of L<sup>A</sup>T<sub>E</sub>X's itemize environment and NFSS declarations.**

In addition to global customization of when to insert the italic correction, it is of course sometimes necessary to explicitly insert one with `\/`.

It is also possible to suppress the italic correction in individual instances. For this, the command `\nocorr` is provided.

The `\nocorr` must appear as the first or last token inside the braces of the argument of the `\text...` commands, at that end of the text where you wish to suppress the italic correction.

It is worth pointing out here that inserting a `\/` in places where it can have no function (i.e. anywhere except immediately after a slanted letter) is not an error—it will just be silently ignored. Unfortunately this is not true if the redefinition of `\/` in `amstex.sty` is used as this version can cause space to be removed immediately before the `\/`.

## 48 The implementation

`\DeclareTextFontCommand` This is the creator function for `\text..` commands. It gives a warning if `\foo` or `\fragfoo` is already defined.

In math mode it simply puts the font declaration and text into a box (possibly an automagically sized one).

Otherwise it first scans the text to see where `\nocorr` occurs within it. This sets the `\check@ic` commands to do what is necessary concerning the italic correction at both ends.

The algorithm for deciding whether to put in an italic correction is not very subtle: one is added whenever the newly current font is not itself positively sloped, unless the next token is a character in the ‘nocorr’ list. At the end of the text this is done after closing the group so as to check the ‘outer font’. Note that this will often result in adding an italic correction token after a character in an unsloped font; we believe (in early 2003) that this is perhaps inefficient but not dangerous.

It also now checks for empty contents of the text command and optimises this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other `aftergroup` tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```

1 (*2kernel)
2 \def \DeclareTextFontCommand #1#2{%
3 \DeclareRobustCommand#1[1]{%
4 \ifmmode
5 \nfss@text{#2##1}%
6 \else
7 \hmode@bgroup

```



```

8 \text@command{##1}%
9 #2\check@icl ##1\check@icr
10 \expandafter
11 \egroup
12 \fi
13 }%
14 }

```

`\textrm` Now we define the `\text{family}` commands in terms of the above; `\texttt` does not look very nice!

```

\texttt 15 \DeclareTextFontCommand{\textrm}{\rmfamily}
\textnormal 16 \DeclareTextFontCommand{\textsf}{\sffamily}
17 \DeclareTextFontCommand{\texttt}{\ttfamily}
18 \DeclareTextFontCommand{\textnormal}{\normalfont}

```

`\textbf` For the series attribute:

```

\textmd 19 \DeclareTextFontCommand{\textbf}{\bfseries}
20 \DeclareTextFontCommand{\textmd}{\mdseries}

```

`\textit` And for the shapes:

```

\textsl 21 \DeclareTextFontCommand{\textit}{\itshape}
\textsc 22 \DeclareTextFontCommand{\textsl}{\slshape}
\textup 23 \DeclareTextFontCommand{\textsc}{\scshape}
24 \DeclareTextFontCommand{\textup}{\upshape}

```

`\emph` Finally we have the `\em` font change declaration of L<sup>A</sup>T<sub>E</sub>X. The corresponding definition with argument is

```
25 \DeclareTextFontCommand{\emph}{\em}
```

`\nocorr` This is just a label, so it does nothing; it should also be unexpandable.

```
26 \let \nocorr \relax
```

`\check@icl` We define these defaults in case some error causes them to be expanded at the wrong time.

```

\check@icr 27 \let \check@icl \@empty
28 \let \check@icr \@empty

```

`\text@command` This checks for a `\nocorr` as the first token in its argument and also for one in any other position not protected within braces (the latter is treated as if it were at the end of the argument).

Is this the correct action in the ‘empty’ case? It is efficient but typographically it is, strictly, incorrect!

```

29 \def \text@command #1{%
30 \def \reserved@a {#1}%
31 \ifx \reserved@a \@empty
32 \let \check@icl \@empty
33 \let \check@icr \@empty
34 \else

```

`\space` is a reserved word in L<sup>A</sup>T<sub>E</sub>X or actually already in plain T<sub>E</sub>X. If somebody really redefines it so many things will break that I don’t see any reason to make this routine here slower than necessary.

```
35 % \def \reserved@b { }%
```

```

36 % \ifx \reserved@a \reserved@b
37 \ifx \reserved@a \space
38 \let \check@icl \@empty
39 \let \check@icr \@empty
40 \else
41 \check@nocorr@ #1\nocorr\@nil
42 \fi
43 \fi
44 }
45 \def \check@nocorr@ #1#2\nocorr#3\@nil {%

```

The two checks are initialised here to their values in the normal case.

```

46 \let \check@icl \maybe@ic
47 \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi}%
48 \def \reserved@a {\nocorr}%
49 \def \reserved@b {#1}%
50 \def \reserved@c {#3}%
51 \ifx \reserved@a \reserved@b
52 \ifx \reserved@c \@empty

```

In this case there is a `\nocorr` at the start but not at the end, so `\check@icl` should be empty.

```

53 \let \check@icl \@empty
54 \else

```

Otherwise there is a `\nocorr` both at the start and elsewhere, so no italic corrections should be added.

```

55 \let \check@icl \@empty
56 \let \check@icr \@empty
57 \fi
58 \else
59 \ifx \reserved@c \@empty

```

In this case there is no `\nocorr` anywhere, so we need to check for an italic correction at both the beginning and the end. This has been set up as the default so no code is needed here.

```

60 \else

```

In this case there is no `\nocorr` at the start but there is one elsewhere, so no `\aftergroup` is needed.

```

61 \let \check@icr \@empty
62 \fi
63 \fi
64 }

```

`\ifmaybe@ic` Switch used solely within `\maybe@ic` not interfering with other switches.

```

65 \newif\ifmaybe@ic

```

`\maybe@ic` These macros implement the italic correction.

```

\maybe@ic@
66 \def \maybe@ic {\futurelet\@let@token\maybe@ic@}
67 \def \maybe@ic@ {%

```

We first check to see if the current font is positively sloped. (But do not forget the message Rainer sent about an upright font with non-zero slope! Or is this an urban myth?) It has been suggested that this should test against a small positive value, but what?

```

68 \ifdim \fontdimen\@ne\font>\z@
69 \else
70 \maybe@ictrue

```

It would be possible, but probably not worthwhile, to continue the forward scan beyond any closing braces.

```

71 \expandafter\@tfor\expandafter\reserved@a\expandafter:\expandafter=%
72 \nocorrlist

```

We have to hide the `\@let@token` in the macro `\t@st@ic` rather than testing it directly in the loop since it might be `\let` to a `\fi` or `\else`, which would result in chaos.

```

73 \do \t@st@ic

```

Frank thinks that the next bit it is inefficient if done after the second change. Chris thinks that most all of this is inefficient for the commonest cases: but that is the price of a cleverer algorithm. It is certainly needed to deal with the use of `\nolinebreak`.

```

74 \ifmaybe@ic \sw@slant \fi
75 \fi
76 }

```

`\t@st@ic` The next token in the input stream is stored in `\@let@token` via a `\let`, the current token from `\nocorrlist` is stored via `\def` in `\reserved@a`. To compare them we have to fiddle around a bit.

If the only things to check were characters then this could be done via an `\if` thus their catcodes would not matter; but this will not work whilst `\futurelet` is used above.

```

77 \def \t@st@ic {%
78 \expandafter\let\expandafter\reserved@b\expandafter=\reserved@a\relax
79 \ifx\reserved@b\@let@token

```

If they are the same we record the fact and jump out of the loop.

```

80 \maybe@icfalse
81 \@break@tfor
82 \fi
83 }

```

`\sw@slant` The definition of the mysterious `\sw@slant` command is as follows.  
`\fix@penalty` 84 \def \sw@slant {%

It is surely correct to put in an italic correction when there is no skip. If the last thing on the list is actually a zero skip (including things whose dimension part is zero, such as `\hfill`), or anything other than a character, then the italic correction will have no effect.

In order to work correctly with unbreakable spaces from `~` (and other common forms of line-breaking control) we also move back across a penalty before the glue.

```

85 \ifdim \lastskip=\z@
86 \fix@penalty
87 \else
88 \skip@ \lastskip
89 \unskip
90 \fix@penalty
91 \hskip \skip@

```

```

92 \fi
93 }

```

The above code means: “If there is a non-zero space just before the current position (`\ifdim...`) save the amount of that space (`\skip@lastskip`), remove it (`\unskip`), then do a similar thing if there is a penalty just before the skip, and finally put the space back in.”

Since zero glue cannot be distinguished in this context from no glue, we dare not put in an `\hskip` in this case as this may produce an unwanted breakpoint. This is not satisfactory.

The penalty before the glue is handled similarly, with the same caveats concerning the zero case. Is this the first recorded use of `\unpenalty` in standard L<sup>A</sup>T<sub>E</sub>X code?

```

94 \def \fix@penalty {%
95 \ifnum \lastpenalty=\z@
96 \@@italiccorr
97 \else
98 \count@ \lastpenalty
99 \unpenalty
100 \@@italiccorr
101 \penalty \count@
102 \fi
103 }

```

`\nocorrlist` This holds the list of characters that should prevent italic correction. They should be ordered by decreasing frequency of use. If any such character is made active later on one needs to redefine the list so that the active character becomes part of it.

```

104 \def \nocorrlist {,.}

```

`\nfss@text` This command will by default behave like a L<sup>A</sup>T<sub>E</sub>X `\mbox` but may be redefined by packages such as `amstext.sty` to be a bit cleverer.

```

105 \ifx \nfss@text\undefined
106 \def \nfss@text {\leavevmode\hbox}
107 \fi

```

`\DeclareOldFontCommand` This is the function used to create declarative font-changing commands that can also be used to change alphabets in math-mode.

Usage: `\DeclareOldFontCommand \fn{<font-change decls>} <math-alphabet>`

Here `\fn` is the font-declaration command being defined, `<font-change decls>` is the declaration it will expand to in text-mode, and `<math-alphabet>` is the (single) math alphabet specifier which is to be used in math-mode.

It does not care whether the command being defined already exists but it does give a warning if it redefines anything.

Here are some typical examples of its use in conjunction with more basic NFSS2 font commands.

```

\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}

```

```

108 \def \DeclareOldFontCommand #1#2#3{%
109 \DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
110 }

```

`\@fontswitch` These two commands actually do the necessary tests and declarative font- or  
`\@math@egroup` alphabet-changing.

```

\@math@egroup
\@math@egroup
111 \def \@fontswitch #1#2{%
112 \ifmmode
113 \let \math@bgroup \relax
114 \def \math@egroup {\let \math@bgroup \@math@bgroup
115 \let \math@egroup \@math@egroup}%

```

We need to have a `\relax` in the following line in case the `#2` is something like `\mathsf` grabbing the next token as an argument. For this reason the code also uses explicit arguments again (see pr/1275).

```

116 #2\relax
117 \else
118 #1%
119 \fi
120 }
121 \let \@math@bgroup \math@bgroup
122 \let \@math@egroup \math@egroup

```

These commands are available only in the preamble.

```

123 \@onlypreamble \DeclareTextFontCommand
124 \@onlypreamble \DeclareOldFontCommand

```

## 49 Initialization

`\normalsize` This is defined to produce an error.

```

125 \def\normalsize{%
126 \@latex@error {The font size command \protect\normalsize\space
127 is not defined:\MessageBreak
128 there is probably something wrong with
129 the class file}\@eha
130 }
131 </2ekernel>

```

# File w ltpageno.dtx

## 50 Page Numbering

Page numbers are produced by a page counter, used just like any other counter. The only difference is that `\c@page` contains the number of the next page to be output (the one currently being produced), rather than one minus it. Thus, it is normally initialized to 1 rather than 0. `\c@page` is defined to be `\count0`, rather than a count assigned by `\newcount`.

`\pagenumbering` The user sets the pagenumber style with the `\pagenumbering{<foo>}` command, which sets the page counter to 1 and defines `\thepage` to be `\foo`. For example, `\pagenumbering{roman}` causes pages to be numbered i, ii, etc.

```
1 (*2kernel)
2 \message{page nos.,}
3 \countdef\c@page=0 \c@page=1
4 \def\c1@page{}
5 \def\pagenumbering#1{%
6 \global\c@page \one \gdef\thepage{\csname @#1\endcsname
7 \c@page}}
8 </2kernel)
```

# File x

## ltxref.dtx

### 51 Cross Referencing

The user writes `\label{foo}` to define the following cross-references:

`\ref{foo}`: value of most recently incremented referencable counter. in the current environment. (Chapter, section, theorem and enumeration counters counters are referencable, footnote counters are not.)

`\pageref{foo}`: page number at which `\label{foo}` command appeared. where foo can be any string of characters not containing ‘\’, ‘{’ or ‘}’.

Note: The scope of the `\label` command is delimited by environments, so `\begin{theorem} \label{foo} ... \end{theorem} \label{bar}` defines `\ref{foo}` to be the theorem number and `\ref{bar}` to be the current section number.

Note: `\label` does the right thing in terms of spacing – i.e., leaving a space on both sides of it is equivalent to leaving a space on either side.

#### 51.1 Cross Referencing

```
1 (*2kernel)
2 \message{x-ref,}
```

This is implemented as follows. A referencable counter `CNT` is incremented by the command `\refstepcounter{CNT}`, which sets `\@currentlabel == {CNT}{eval(\p@cnt\theCNT)}`. The command `\label{FOO}` then writes the following on file `\@auxout`:

```
\newlabel{FOO}{{eval(\@currentlabel)}{eval(\thepage)}}
```

```
\ref{FOO} ==
BEGIN
 if \r@foo undefined
 then @refundefined := G T
 ??
 Warning: 'reference foo on page ... undefined'
 else \@car \eval(\r@FOO)\@nil
 fi
END
```

```
\pageref{foo} =
BEGIN
 if \r@foo undefined
 then @refundefined := G T
 ??
 Warning: 'reference foo on page ... undefined'
 else \@cdr \eval(\r@FOO)\@nil
 fi
END
```

`\G@refundefinedtrue` This does not save on name-space (since `\G@refundefinedfalse` was never needed) but it does make the implementation of such one-way switches more consistent. The extra macro to make the change is used since this change appears several times.

**Note** despite its name, `\G@refundefinedtrue` does *not* correspond to an `\if` command, and there is no matching `...false`. It would be more natural to call the command `\G@refundefined` (as inspection of the change log will reveal) but unfortunately such a change would break any package that had defined a `\ref`-like command that mimicked the definition of `\ref`, calling `\G@refundefinedtrue`. Inspection of the T<sub>E</sub>X archives revealed several such packages, and so this command has been named `...true` so that the definition of `\ref` need not be changed, and the packages will work without change.

```

3 % \newif\ifG@refundefined
4 % \def\G@refundefinedtrue{\global\let\ifG@refundefined\iftrue}
5 % \def\G@refundefinedfalse{\global\let\ifG@refundefined\iffalse}
6 \def\G@refundefinedtrue{%
7 \gdef\@refundefined{%
8 \@latex@warning@no@line{There were undefined references}}
9 \let\@refundefined\relax

```

`\ref` Referencing a `\label`. RmS 91/10/25: added a few extra `\reset@font`, as suggested by Bernd Raichle

`\pageref` RmS 92/08/14: made `\ref` and `\pageref` robust  
`\setref` RmS 93/09/08: Added setting of `refundefined` switch.

```

10 \def\@setref#1#2#3{%
11 \ifx#1\relax
12 \protect\G@refundefinedtrue
13 \nfss@text{\reset@font\bfseries ??}%
14 \@latex@warning{Reference ‘#3’ on page \thepage \space
15 undefined}%
16 \else
17 \expandafter#2#1\null
18 \fi}
19 \def\ref#1{\expandafter\@setref\c@name r@#1\endc@name\@firstoftwo{#1}}
20 \def\pageref#1{\expandafter\@setref\c@name r@#1\endc@name
21 \@secondoftwo{#1}}

```

`\newlabel` This command will be written to the `.aux` file to pass label information from one run to another.

`\@newl@bel` The internal form of `\newlabel` and `\bibcite`. Note that this macro does its work inside a group. That way the local assignments it needs to do don't clutter the save stack. This prevents large documents with many labels to run out of save stack.

```

22 \def\@newl@bel#1#2#3{%
23 \@ifundefined{#1@#2}%
24 \relax
25 {\gdef \@multiplelabels {%
26 \@latex@warning@no@line{There were multiply-defined labels}}%
27 \@latex@warning@no@line{Label ‘#2’ multiply defined}}%
28 \global\@namedef{#1@#2}{#3}}

```



```

29 \def\newlabel{\@newl@bel r}
30 \@onlypreamble\@newl@bel

\if@multiplelabels This is redefined to produce a warning if at least one label is defined more than
\@multiplelabels once. It is executed by the \enddocument command.
31 \let \@multiplelabels \relax

\label The commands \label and \refstepcounter have been changed to allow
\refstepcounter \protect'ed commands to work properly. For example,

\def\thechapter{\protect\foo{\arabic{chapter}.\roman{section}}}

will cause a \label{bar} command to define \ref{bar} to expand to something
like \foo{4.d}. Change made 20 Jul 88.

32 \def\label#1{\@bsphack
33 \protected@write\@auxout{}%
34 {\string\newlabel{#1}{\@currentlabel}{\thepage}}%
35 \@esphack}

36 \def\refstepcounter#1{\stepcounter{#1}%
37 \protected@edef\@currentlabel
38 {\csname p@#1\endcsname\csname the#1\endcsname}%
39 }

\@currentlabel For \label commands that come before any environment
40 \def\@currentlabel{}

41 </2kernel)

```

## 51.2 An extension of counter referencing

At the moment a reference to a counter `foo` will generate the equivalent of `\p@foo\thefoo` although not quite in this form. For some applications it would be nice if one could have `\thefoo` being an argument to `\p@foo` to be able to put material before and after the number generated by `\thefoo`. This can be easily achieved with a small change to one of the kernel commands as follows:

```

\def\refstepcounter#1{\stepcounter{#1}%
\protected@edef\@currentlabel
{\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}

```

The trick is to ensure that `\csname the#1\endcsname` is turned into a single token before `\p@...` is expanded further. This way, if the `\p@...` command is a macro with one argument it will receive `\the...`. With the kernel code (i.e., without the `\expandafter`) it will instead pick up `\csname` which would be disastrous.

Using `\expandafter` instead of braces delimiting the argument is better because, assuming that the `\p@...` command is not defined as a macro with one argument, the braces will stay and prohibit kerning that might otherwise happen between the glyphs generated by `\the...` and surrounding glyphs.

We have refrained from making this change in the kernel code although for existing documents it would be 100% backward compatible. The reason being

that any class or package making use of this functionality would then horribly fail with older L<sup>A</sup>T<sub>E</sub>X installations.

Instead we suggest that people who are interested in using this functionality in a document class or package add the redefinition to the class file. To ensure that this redefinition is properly applied they might want to test for the original definition first, e.g.

```
\CheckCommand*\refstepcounter[1]{\stepcounter{#1}%
 \protected@edef\@currentlabel
 {\csname p@#1\endcsname\csname the#1\endcsname}%
}
\renewcommand*\refstepcounter[1]{\stepcounter{#1}%
 \protected@edef\@currentlabel
 {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```

# File y

## ltmiscen.dtx

### 52 Miscellaneous Environments

This section implements the basic environment mechanism, and also a few specific environments including `document`, The math environments and related commands, the ‘flushing’ environments, (`center`, `flushleft`, `flushright`), and `verbatim`.

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

#### 52.1 Environments

`\begin{foo}` and `\end{foo}` are used to delimit environment `foo`.

`\begin{foo}` starts a group and calls `\foo` if it is defined, otherwise it does nothing.

`\end{foo}` checks to see that it matches the corresponding `\begin` and if so, it calls `\endfoo` and does an `\endgroup`. Otherwise, `\end{foo}` does nothing.

If `\end{foo}` needs to ignore blanks after it, then `\endfoo` should globally set the `@ignore` switch true with `\@ignoretrue` (this will automatically be global).

NOTE: `\@@end` is defined to be the `\end` command of T<sub>E</sub>X82.

`\enddocument` is the user’s command for ending the manuscript file.

`\stop` is a panic button — to end T<sub>E</sub>X in the middle.

```
\enddocument ==
BEGIN
 \@checkend{document} %% checks for unmatched \begin
 \clearpage
 \begingroup
 if @filesw = true
 then close file @mainaux
 if G@refundefined = true
 then LaTeX Warning: 'There are undefined references.' fi
 if @multiplelabels = true
 then LaTeX Warning:
 'One or more label(s) multiply defined.'
 else
 \@setckpt {ARG1}{ARG2} == null
 \newlabel{LABEL}{VAL} ==
 BEGIN
 \reserved@a == VAL
 if def(\reserved@a) = def(\r@LABEL)
 else @tempswa := true fi
 END
 \bible{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

```

`\@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 environments 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 \@tempwatrue \fi}
```

```
\@writefile
```

```
43 \long\def\@writefile#1#2{%
44 \ifundefined{tf#1}\relax
45 {\@temptokena{#2}%
46 \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
47 }%
48 }
```

```

\stop
49 \def\stop{\clearpage\deadcycles\z@\let\par\@par\@end}

50 \everypar{\@nodocument} %% To get an error if text appears before the
51 \nullfont %% \begin{document}

\begin, \end, and \@checkend changed so \end{document} will catch
an unmatched \begin. Changed 24 May 89 as suggested by
Frank Mittelbach and Rainer Sch\"opf.

\begin{NAME} ==
BEGIN
 IF \NAME undefined THEN \reserved@a == BEGIN report error
END
 ELSE \reserved@a ==
 (\@currentvir :=L NAME) \NAME
 FI
 @ignore :=G F %% Added 30 Nov 88
 \begingroup
 \@endpe := F
 \@currentvir :=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
 FI %% immediately following text
 IF @ignore = T
 THEN @ignore :=G F
 \ignorespaces
 FI
END

\@checkend{NAME} ==
BEGIN
 IF \@currentvir = NAME
 ELSE \@badend{NAME}
 FI
END

```

```

\begin
52 \def\begin#1{%
53 \ifundefined{#1}%
54 {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
55 {\def\reserved@a{\def\@currenvir{#1}%
56 \edef\@currencline{\on@line}%
57 \csname #1\endcsname}}%
58 \@ignorefalse
59 \begingroup\@endpfalse\reserved@a}

```

```

\end
60 \def\end#1{%
61 \csname end#1\endcsname\@checkend{#1}%
62 \expandafter\endgroup\if@endpe\@doendpe\fi
63 \if@ignore\@ignorefalse\ignorespaces\fi}

```

```

\@checkend
64 \def\@checkend#1{\def\reserved@a{#1}\ifx
65 \reserved@a\@currenvir \else\@badend{#1}\fi}

```

`\@currencline` We do need a default value for `\@currencline` on top-level since the document environment cancels the brace group. This means that a mismatch with `\begin{document}` will not produce a line number. Thus the outer default must be `\@empty` or we will end up with two spaces.

```
66 \let\@currencline\@empty
```

## 52.2 Center, Flushright, Flushleft

```
67 \message{center,}
```

```

\center, \flushright and \flushleft set
 \rightskip = 0pt or \@flushglue (as appropriate)
 \leftskip = 0pt or \@flushglue (as appropriate)
 \parindent = 0pt
 \parfillskip = 0pt. (except \flushleft)
 \l == \par \vskip -\parskip
 \l[LENGTH] == \l \vskip LENGTH
 \l* == \par \penalty 10000 \vskip -\parskip
 \l*[LEN] == \l* \vskip LENGTH

```

They invoke the `trivlist` environment to handle vertical spacing before and after them.

`\centering`, `\raggedright` and `\raggedleft` are the declaration analogs of the above.

`\raggedright` has a more universal effect, however. It sets `\@rightskip := flushglue`. Every environment, like the list environments, that set `\rightskip` to its 'normal' value set it to `\@rightskip`

```

\@centercr
68 \def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
69 \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}

\@xcentercr
70 \def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
71 [\@icentercr\ignorespaces}

\@icentercr
72 \def\@icentercr[#1]{\vskip #1\ignorespaces}

center We use \relax to prevent \item scanning too far.
73 \def\center{\trivlist \centering\item\relax}
74 \def\endcenter{\endtrivlist}

\centering
75 \def\centering{%
76 \let\\\@centercr
77 \rightskip\@flushglue\leftskip\@flushglue
78 \parindent\z@\parfillskip\z@skip}

\@rightskip
79 \newskip\@rightskip \@rightskip \z@skip

flushleft We use \relax to prevent \item scanning too far.
80 \def\flushleft{\trivlist \raggedright\item\relax}
81 \def\endflushleft{\endtrivlist}

\raggedright
82 \def\raggedright{%
83 \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
84 \leftskip\z@skip
85 \parindent\z@}

flushright We use \relax to prevent \item scanning too far.
86 \def\flushright{\trivlist \raggedleft\item\relax}
87 \def\endflushright{\endtrivlist}

\raggedleft
88 \def\raggedleft{%
89 \let\\\@centercr
90 \rightskip\z@skip\leftskip\@flushglue
91 \parindent\z@\parfillskip\z@skip}

92 \message{verbatim,}

```



## 52.3 Verbatim

The verbatim environment uses the fixed-width `\ttfamily` font, turns blanks into spaces, starts a new line for each carriage return (or sequence of consecutive carriage returns), and interprets *every* character literally. I.e., all special characters `\`, `{`, `$`, etc. are `\catcode'd` to 'other'.

The command `\verb` produces in-line verbatim text, where the argument is delimited by any pair of characters. E.g., `\verb #...#` takes '...' as its argument, and sets it verbatim in `\ttfamily` font.

The \*-variants of these commands are the same, except that spaces print as the T<sub>E</sub>Xbook's space character instead of as blank spaces.

```
\@vobeyspaces
93 {\catcode'\ =\active%
94 \gdef\@vobeyspaces{\catcode'\ \active\let \@xobeysp}}

\@xobeysp

\@xverbatim
\@sxverbatim
95 \begingroup \catcode '|=0 \catcode '['= 1
96 \catcode']=2 \catcode '\{=12 \catcode '\}=12
97 \catcode'\}=12 |gdef|@xverbatim#1\end{verbatim}#[#1\end[verbatim]]
98 |gdef|@sxverbatim#1\end{verbatim*}#[#1\end[verbatim*]]
99 |endgroup

\@verbatim Real start of verbatim environment We use \relax to prevent \item scanning too
far.
100 \def\@verbatim{\trivlist \item\relax
101 \if@minipage\else\vskip\parskip\fi
102 \leftskip\@totalleftmargin\rightskip\z@skip
103 \parindent\z@\parfillskip\@flushglue\parskip\z@skip
Added \@@par to clear possible \parshape definition from a surrounding list (the
verbatim guru says).
104 \@@par
105 \@tempwafalse
106 \def\par{%
107 \if@tempswa
A \leavevmode added: needed if, for example, a blank verbatim line is the first
thing in a list item (wow!).
108 \leavevmode \null \@@par\penalty\interlinepenalty
109 \else
110 \@tempswatrue
111 \ifhmode\@@par\penalty\interlinepenalty\fi
112 \fi}%
To allow customization we hide the font used in a separate macro.
113 \let\do\@makeother \dospecials
114 \obeylines \verbatim@font \@noligs
115 \hyphenchar\font\@m@ne
To avoid a breakpoint after the labels box, we remove the penalty put there by
the list macros: another use of \unpenalty!
116 \everypar \expandafter{\the\everypar \unpenalty}%
117 }
```

```

\verbatim (RmS 93/09/19) Protected against 'missing item' error message triggered by
\endverbatim empty verbatim environment.
118 \def\verbatim{\@verbatim \frenchspacing\@vobeyspaces \@xverbatim}
119 \def\endverbatim{\if@newlist \leavevmode\fi\endtrivlist}

\verbatim@font Macro to select the font used for verbatim typesetting. It also does other work if
necessary for the font used.
120 \def\verbatim@font{\normalfont\ttfamily}

verbatim*
121 \@namedef{verbatim*}{\@verbatim\@sxverbatim}
122 \expandafter\let\csname endverbatim*\endcsname =\endverbatim

\@makeother
123 \def\@makeother#1{\catcode'#112\relax}

\verb@balance@group
124 \let\verb@balance@group\@empty

\verb@egroup
125 \def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}

\verb@eol@error
126 \begingroup
127 \obeylines%
128 \gdef\verb@eol@error{\obeylines%
129 \def~M{\verb@egroup\@latex@error{%
130 \noexpand\verb ended by end of line}\@ehc}}%
131 \endgroup

\verb Typesetting a small piece verbatim.
132 \def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
133 \bgroup
134 \verb@eol@error \let\do\@makeother \dospecials
135 \verbatim@font\@noligs
136 \@ifstar\@sverb\@verb}

\@sverb Definitions of \@sverb and \@verb changed so \verb+ foo+ does not lose lead-
ing blanks when it comes at the beginning of a line. Change made 24 May 89.
Suggested by Frank Mittelbach and Rainer Schöpf.
137 \def\@sverb#1{%
138 \catcode'#1\active
139 \lccode'\~'#1%
140 \gdef\verb@balance@group{\verb@egroup
141 \@latex@error{\noexpand\verb illegal in command argument}\@ehc}%
142 \aftergroup\verb@balance@group
143 \lowercase{\let~\verb@egroup}}%

\@verb
144 \def\@verb{\@vobeyspaces \frenchspacing \@sverb}

\verbatim@nolig@list
145 \def\verbatim@nolig@list{\do\~\do\<\do\>\do\,\do\'\do\~}

```

```

\do@noligs
146 \def\do@noligs#1{%
147 \catcode'#1\active
148 \begingroup
149 \lccode'\~'#1\relax
150 \lowercase{\endgroup\def~{\leavevmode\kern\z@\char'#1}}}}

\@noligs To stay compatible with packages that use \@noligs we keep it.
151 \def\@noligs{\let\do\do@noligs \verbatim@nolig@list}

152 </2kernel)

```

## File z

# ltmath.dtx

## 53 Math setup

This file contains a lot of the original plain T<sub>E</sub>X code, as well as the L<sup>A</sup>T<sub>E</sub>X environments for math. It still needs sorting out.

```
1 (*2kernel)
2 \message{math definitions,}
```

### 53.1 Math commands based on plain T<sub>E</sub>X

#### 53.1.1 The log-like functions

```
\log The standard operators:
3 \def\log{\mathop{\operator@font log}\nolimits}
4 \def\lg{\mathop{\operator@font lg}\nolimits}
5 \def\ln{\mathop{\operator@font ln}\nolimits}
6 \def\lim{\mathop{\operator@font lim}}
7 \def\limsup{\mathop{\operator@font lim\,sup}}
8 \def\liminf{\mathop{\operator@font lim\,inf}}
9 \def\sin{\mathop{\operator@font sin}\nolimits}
10 \def\arcsin{\mathop{\operator@font arcsin}\nolimits}
11 \def\sinh{\mathop{\operator@font sinh}\nolimits}
12 \def\cos{\mathop{\operator@font cos}\nolimits}
13 \def\arccos{\mathop{\operator@font arccos}\nolimits}
14 \def\cosh{\mathop{\operator@font cosh}\nolimits}
15 \def\tan{\mathop{\operator@font tan}\nolimits}
16 \def\arctan{\mathop{\operator@font arctan}\nolimits}
17 \def\tanh{\mathop{\operator@font tanh}\nolimits}
18 \def\cot{\mathop{\operator@font cot}\nolimits}
19 \def\coth{\mathop{\operator@font coth}\nolimits}
20 \def\sec{\mathop{\operator@font sec}\nolimits}
21 \def\csc{\mathop{\operator@font csc}\nolimits}
22 \def\max{\mathop{\operator@font max}}
23 \def\min{\mathop{\operator@font min}}
24 \def\sup{\mathop{\operator@font sup}}
25 \def\inf{\mathop{\operator@font inf}}
26 \def\arg{\mathop{\operator@font arg}\nolimits}
27 \def\ker{\mathop{\operator@font ker}\nolimits}
28 \def\dim{\mathop{\operator@font dim}\nolimits}
29 \def\hom{\mathop{\operator@font hom}\nolimits}
30 \def\det{\mathop{\operator@font det}}
31 \def\exp{\mathop{\operator@font exp}\nolimits}
32 \def\Pr{\mathop{\operator@font Pr}}
33 \def\gcd{\mathop{\operator@font gcd}}
34 \def\deg{\mathop{\operator@font deg}\nolimits}

\bmod And some operators have to be done by hand:
35 \def\bmod{%
36 \nonscript\mskip-\medmuskip\mkern5mu%
```

```

37 \mathbin{\operator@font mod}\penalty900\mkern5mu%
38 \nonscript\mskip-\medmuskip}

```

`\pmod`

```

39 \def\pmod#1{%
40 \allowbreak\mkern18mu({\operator@font mod}\,\,#1)}

```

### 53.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```

41 \def\bigl{\mathopen\big}
42 \def\bigm{\mathrel\big}
43 \def\biggr{\mathclose\big}
44 \def\Bigl{\mathopen\Big}
45 \def\Bigm{\mathrel\Big}
46 \def\Biggr{\mathclose\Big}
47 \def\biggl{\mathopen\bigg}
48 \def\biggm{\mathrel\bigg}
49 \def\biggr{\mathclose\bigg}
50 \def\Biggl{\mathopen\Bigg}
51 \def\Biggm{\mathrel\Bigg}
52 \def\Biggr{\mathclose\Bigg}

```

### 53.1.3 The UNSORTED Rest

The other math commands are lifted from plain T<sub>E</sub>X.

`\jot`

```

53 \newdimen\jot
54 \jot=3pt

```

`\interdisplaylinepenalty`

```

55 \newcount\interdisplaylinepenalty
56 \interdisplaylinepenalty=100

```

`\choose`

```

57 \def\choose{\atopwithdelims()}

```

`\brack`

```

58 \def\brack{\atopwithdelims[]}

```

`\brace`

```

59 \def\brace{\atopwithdelims\{\}}

```

`\mathpalette`

```

60 \def\mathpalette#1#2{%
61 \mathchoice
62 {#1\displaystyle{#2}}%
63 {#1\textstyle{#2}}%
64 {#1\scriptstyle{#2}}%
65 {#1\scriptscriptstyle{#2}}

```

```

\root
\rootbox 66 \newbox\rootbox
\root 67 \def\root#1\of{%
\mathpalette\root@#1\of{%
68 \setbox\rootbox\hbox{\m@th\scriptscriptstyle{#1}$}%
69 \mathpalette\root@#1\of{
70 \def\root@#1#2{%
71 \setbox\z@\hbox{\m@th#1\sqrt{#2}}%
72 \dimen@ \ht\z@ \advance\dimen@-\dp\z@
73 \mkern5mu\raise.6\dimen@\copy\rootbox
74 \mkern-10mu\box\z@}

\phantom
\hphantom 75 \newif\ifv@
\vphantom 76 \newif\ifh@

77 \def\vphantom{\v@true\h@false\ph@nt}
78 \def\hphantom{\v@false\h@true\ph@nt}
79 \def

80 \def\ph@nt{%
81 \ifmmode
82 \expandafter\mathpalette\expandafter\mathph@nt
83 \else
84 \expandafter\makeph@nt
85 \fi}

86 \def\makeph@nt#1{%
87 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finph@nt}

88 \def\mathph@nt#1#2{%
89 \setbox\z@\hbox{\m@th#1{#2}}\finph@nt}

90 \def\finph@nt{%
91 \setbox\zw@ \null
92 \ifv@ \ht\zw@ \ht\z@ \dp\zw@ \dp\z@ \fi
93 \ifh@ \wd\zw@ \wd\z@ \fi \box\zw@}

\mathstrut
94 \def\mathstrut{\vphantom{}}

\smash
95 \def\smash{%
96 \relax % \relax, in case this comes first in \halign
97 \ifmmode
98 \expandafter\mathpalette\expandafter\mathsm@sh
99 \else
100 \expandafter\makesm@sh
101 \fi}

102 \def\makesm@sh#1{%
103 \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
104 \def\mathsm@sh#1#2{%
105 \setbox\z@\hbox{\m@th#1{#2}}\finsm@sh}
106 \def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}

```

```

\buildrel
107 \def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}

\cases
108 \def\cases#1{\left\{\,\vcenter{\normalbaselines\m@th
109 \ialign{###\hfil$&\quad{##}\hfil\crrc#1\crrc}}\right.}

\matrix
110 \def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
111 \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crrc
112 \mathstrut\crrc\noalign{\kern-\baselineskip}
113 #1\crrc\mathstrut\crrc\noalign{\kern-\baselineskip}}\,}

\pmatrix
114 \def\pmatrix#1{\left(\matrix{#1}\right)}

\bordermatrix
115 \def\bordermatrix#1{\begingroup \m@th
116 \@tempdima 8.75\p@
117 \setbox\z@\vbox{%
118 \def\cr{\crrc\noalign{\kern2\p@\global\let\cr\endline}}%
119 \ialign{###\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
120 &&\quad\hfil$##$\hfil\crrc
121 \omit\strut\hfil\crrc\noalign{\kern-\baselineskip}%
122 #1\crrc\omit\strut\cr}}%
123 \setbox\tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
124 \setbox\tw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
125 \setbox\tw@\hbox{\kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
126 \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
127 \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,\right)$}%
128 \null;\vbox{\kern\ht\@ne\box\tw@}\endgroup}

\openup
129 \def\openup{\afterassignment\@penup\dimen@}
130 \def\@penup{\advance\lineskip\dimen@
131 \advance\baselineskip\dimen@
132 \advance\lineskiplimit\dimen@}

\displaylines
133 \newif\ifdt@p
134 \def\displ@y{\global\dt@ptrue\openup\jot\m@th
135 \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
136 \vskip-\lineskiplimit \vskip\normallineskiplimit \fi
137 \else \penalty\interdisplaylinepenalty \fi}}}
138 \def\@lign{\tabskip\z@skip\everycr{}} % restore inside \displ@y
139 \def\displaylines#1{\displ@y \tabskip\z@skip
140 \halign{\hb@xt@\displaywidth{\@lign\hfil\displaystyle##\hfil$}\crrc
141 #1\crrc}}

\sp
\sb
142 \let\sp=^
143 \let\sb=_

```

```

\>
\; 144 %\def\,\{\mskip\thinmuskip} % already defined in ltspace
\! 145 \def\>{\mskip\medmuskip}
146 \def\;{\mskip\thickmuskip}
147 \def\!{\mskip-\thinmuskip}

*
148 \def*{\discretionary{\thinspace\the\textfont2\char2}{\{}}{\}}

\: Nickname for the medium space since \> is not available inside tabbing.
149 \let\:=\>

\active@math@prime This is the definition of the active math prime.
150 \def\active@math@prime{\bgroup\prim@s}

\prime@s
151 {\catcode'\='=active \global\let'\active@math@prime}
152 \def\prim@s{%
153 \prime\futurelet\@let@token\pr@m@s}
154 \def\pr@m@s{%
155 \ifx'\@let@token
156 \expandafter\pr@@@s
157 \else
158 \ifx^\@let@token
159 \expandafter\expandafter\expandafter\pr@@@t
160 \else
161 \egroup
162 \fi
163 \fi}
164 \def\pr@@@s#1{\prim@s}
165 \def\pr@@@t#1#2{#2\egroup}

166 {\catcode'_ =active \gdef_{_}} % _ in math is
167 % either subscript or _

```

## 53.2 Math Environments

```

\(

\(

168 </2ekernel>
169 <(latexrelease)\IncludeInRelease{2015/01/01}{\{()\{Make \(\ robust}}%
170 <(*2ekernel | latexrelease)>
171 \DeclareRobustCommand\(

172 \relax\ifmmode\@badmath\else$\fi}%
173 \DeclareRobustCommand\(

174 \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
175 </2ekernel | latexrelease>
176 <(latexrelease)\EndIncludeInRelease
177 <(latexrelease)\IncludeInRelease{0000/00/00}{\{()\{Make \(\ robust}}%
178 <(latexrelease)\def\(


```



```

179 <latexrelease> \relax\ifmmode\@badmath\else$\fi}%
180 <latexrelease>\def\}%
181 <latexrelease> \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
182 <latexrelease>\EndIncludeInRelease
183 <*/2ekernel>

```

\[ 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  $$$$ ).

```

184 </2ekernel>
185 <latexrelease>\IncludeInRelease{2015/01/01}{\[\]{Make \[robust}}%
186 <*/2ekernel | latexrelease>
187 \DeclareRobustCommand\[%
188 \relax\ifmmode
189 \@badmath
190 \else
191 \ifvmode
192 \nointerlineskip
193 \makebox[.6\linewidth]{}%
194 \fi
195 $$$%$$ BRACE MATCH HACK
196 \fi
197 }%

198 \DeclareRobustCommand\}%
199 \relax\ifmmode
200 \ifinner
201 \@badmath
202 \else
203 $$$%$$ BRACE MATCH HACK
204 \fi
205 \else
206 \@badmath
207 \fi
208 \ignorespaces
209 }%

210 </2ekernel | latexrelease>
211 <latexrelease>\EndIncludeInRelease
212 <latexrelease>\IncludeInRelease{0000/00/00}{\[\]{Make \[robust}}%
213 <latexrelease>\def\[%
214 <latexrelease> \relax\ifmmode
215 <latexrelease> \@badmath
216 <latexrelease> \else
217 <latexrelease> \ifvmode
218 <latexrelease> \nointerlineskip
219 <latexrelease> \makebox[.6\linewidth]{}%
220 <latexrelease> \fi
221 <latexrelease> $$$%$$ BRACE MATCH HACK
222 <latexrelease> \fi
223 <latexrelease>}%

224 <latexrelease>\def\}%
225 <latexrelease> \relax\ifmmode
226 <latexrelease> \ifinner
227 <latexrelease> \@badmath

```

```

228 (latexrelease) \else
229 (latexrelease) $$$%$$$ BRACE MATCH HACK
230 (latexrelease) \fi
231 (latexrelease) \else
232 (latexrelease) \@badmath
233 (latexrelease) \fi
234 (latexrelease) \ignorespaces
235 (latexrelease)}%
236 (latexrelease)\EndIncludeInRelease
237 (*2kernel)

math Disguises for \(...\) and \[... \].
displaymath 238 \let\math=\(
239 \let\endmath=\)

240 \def\displaymath{\[}
241 \def\enddisplaymath{\]\@ignoretrue}

equation Numbered equations, using the counter \c@equation. Note: The document style
\c@equation must define \theequation etc., and do the appropriate \@addtoreset. It should
also redefine \@eqnnum if another format for the equation number is desired other
than the standard (...), or to move the equation numbers to the flushleft. (See
comment on the \def of \@eqnnum.)

242 \@definecounter{equation}
243 \def\equation{$$\refstepcounter{equation}}
244 \def\endequation{\eqno \hbox{\@eqnnum}$$\@ignoretrue}

\@eqnnum Produces the equation number for equation and eqnarray environments. The
following definition is for flushright numbers; for flushleft numbers, see leqno.clo.
The equation number is set in black roman type even if an eqnarray environment
appears in an italic environment.

245 \def\@eqnnum{\normalfont \normalcolor (\theequation)}

\stackrel A disguise for plain TEX's buildrel.

246 \def\stackrel#1#2{\mathrel{\mathop{#2}\limits^{#1}}}

\frac A disguise for plain TEX's \over.

247 \def\frac#1#2{\begingroup#1\endgroup\over#2}

\sqrt Add an optional argument to plain's \sqrt to give the nth root of an expression
\@sqrt $\sqrt[n]{e}$.

248 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign]}
249 \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 250 \newcount\@eqcnt
\if@eqnsw 251 \newcount\@eqpen
\@eqnset 252 \newif\if@eqnsw\@eqnswtrue
253 \newskip\@centering
254 \@centering = 0pt plus 1000pt

```

To get a proper `\@currentlabel` we have to redefine it for the whole display. Note that we can't use `\refstepcounter` as this results in `\@currentlabel` getting restored at the wrong and thus always writing the first label to the `.aux` file.

```

255 \def\eqnarray{%
256 \stepcounter{equation}%
257 \def\@currentlabel{\p@equation\theequation}%
258 \global\@eqnswtrue
259 \m@th
260 \global\@eqcnt\z@
261 \tabskip\@centering
262 \let\\\@eqncr
263 $$\everycr{\halign to\displaywidth\bgroup
264 \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnset
265 &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil$##$\hfil
266 &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
267 $\displaystyle{##}$\hfil\tabskip\@centering
268 &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
269 \tabskip\z@skip
270 \cr
271 }

272 \def\endeqnarray{%
273 \@eqncr
274 \egroup
275 \global\advance\c@equation\m@ne
276 $$\@ignoretrue
277 }

278 \let\@eqnset=\relax

```

`\nonumber` Switches off equation numbering.

```

279 \def\nonumber{\global\@eqnswfalse}

```

`\@eqncr`

```

\@xeqncr 280 \def\@eqncr{%
\@yeqncr 281 {\ifnum0='}\fi
282 \@ifstar{%
283 \global\@eqpen\@M\@yeqncr
284 }{%
285 \global\@eqpen\interdisplaylinepenalty \@yeqncr
286 }%
287 }

288 \def\@yeqncr{\@testopt\@xeqncr\z@skip}

289 \def\@xeqncr[#1]{%
290 \ifnum0='{ \fi}%
291 \@@eqncr
292 \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
293 }

```

`\@@eqncr`

```

294 \def\@@eqncr{\let\reserved@a\relax
295 \ifcase\@eqcnt \def\reserved@a{& &} \or \def\reserved@a{& &}%
296 \or \def\reserved@a{&} \else
297 \let\reserved@a\@empty

```

```

298 \@latex@error{Too many columns in eqnarray environment}\@ehc\fi
299 \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
300 \global\@eqnswtrue\global\@eqcnt\z@\cr}

```

`eqnarray*` Here's the `eqnarray*` environment:

```

\@seqnucr 301 \let\@seqnucr=\@eqnucr
 302 \@namedef{eqnarray*}{\def\@eqnucr{\nonumber\@seqnucr}\eqnarray}
 303 \@namedef{endeqnarray*}{\nonumber\endeqnarray}

```

`\lefteqn` `\lefteqn{FORMULA}` typesets FORMULA in display math style flushleft in a box of width zero.

```

304 \def\lefteqn#1{\rlap{\$}\displaystyle #1$}

```

`\ensuremath` In math mode, `\ensuremath{text}` is equivalent to `text`; in LR or paragraph mode, it is equivalent to `$text$`. `\relax` is not needed in front of the `\ifmmode` as `\protect` will be `\let` to `\relax`. This version (due to Donald Arseneau) avoids duplicating its argument in the 'then' and 'else' part of the `\ifmath` which is necessary in nested 'tabular' like environments. See `amslatex/2104`.

```

305 \DeclareRobustCommand{\ensuremath}{%
306 \ifmmode
307 \expandafter\@firstofone
308 \else
309 \expandafter\@ensuredmath
310 \fi}

```

`\@ensuredmath` The `\relax` stops `\ensuremath{}` starting display math.

```

311 \long\def\@ensuredmath#1{\relax#1$}
312 </2kernel>

```

## 53.3 External options to the standard document classes

### 53.3.1 Left equation numbering

`\@eqnnum` To put the equation number on the left side of an equation we have to use a little trick. The number is shifted `\displaywidth` to the left inside a box of (approximately) zero width. This fails when the equation is too wide, the equation number than may overprint the equation itself.

```

313 <*\leqno>
314 \renewcommand\@eqnnum{\hb@xt@.01\p@{}%
315 \rlap{\normalfont\normalcolor
316 \hskip -\displaywidth(\theequation)}}
317 </\leqno>

```

### 53.3.2 Flush left equations

To get the displayed math environments to print the contents flush left (with an indentation) we have to redefine all of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s displayed math environments.

`\mathindent` The amount of indentation of the equations is stored in a register.

```

318 <*\fleqn>
319 \newdimen\mathindent

```

The setting of `\mathindent` has to be deferred until the class file has been processed, because `\leftmargin` is still 0pt wide at the moment `fleqn.clo` is read in.

```
320 \AtEndOfClass{\mathindent\leftmargin}
```

`\[` Begin display math;

```
321 \IncludeInRelease{2015/01/01}{\[]}{Make \[robust}%
322 \DeclareRobustCommand\[\relax
323 \ifmmode\@badmath
324 \else
325 \begin{trivlist}%
326 \@beginparpenalty\predisplaypenalty
327 \@endparpenalty\postdisplaypenalty
328 \item[]\leavevmode
329 \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
330 \hskip\mathindent\bgroup
331 \fi}
332 \EndIncludeInRelease

333 \IncludeInRelease{0000/00/00}{\[]}{Make \[robust}%
334 \renewcommand\[\relax
335 \ifmmode\@badmath
336 \else
337 \begin{trivlist}%
338 \@beginparpenalty\predisplaypenalty
339 \@endparpenalty\postdisplaypenalty
340 \item[]\leavevmode
341 \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
342 \hskip\mathindent\bgroup
343 \fi}
344 \EndIncludeInRelease
```

`\]` end display math;

```
345 \IncludeInRelease{2015/01/01}{\}\]{Make \] robust}%
346 \DeclareRobustCommand\]\relax
347 \ifmmode
348 \egroup $\hfil% $
349 \egroup
350 \end{trivlist}%
351 \else \@badmath
352 \fi}
353 \EndIncludeInRelease

354 \IncludeInRelease{0000/00/00}{\}\]{Make \] robust}%
355 \renewcommand\]\relax
356 \ifmmode
357 \egroup $\hfil% $
358 \egroup
359 \end{trivlist}%
360 \else \@badmath
361 \fi}
362 \EndIncludeInRelease
```

`equation` The equation environment

```

363 \renewenvironment{equation}%
364 {\@beginparpenalty\predisplaypenalty
365 \@endparpenalty\postdisplaypenalty
366 \refstepcounter{equation}%
367 \trivlist \item[]\leavevmode
368 \hb@xt@\linewidth\bgroup $\m@th% $
369 \displaystyle
370 \hskip\mathindent}%
371 {${\hfil % $
372 \displaywidth\linewidth\hbox{\@eqnum}%
373 \egroup
374 \endtrivlist}

```

eqnarray The eqnarray environment

```

375 \renewenvironment{eqnarray}{%
376 \stepcounter{equation}%
377 \def\@currentlabel{\p@equation\theequation}%
378 \global\@eqnswtrue\m@th
379 \global\@eqcnt\z@
380 \tabskip\mathindent
381 \let\@=\@eqnocr
382 \setlength\abovedisplayskip{\topsep}%
383 \ifvmode
384 \addtolength\abovedisplayskip{\partopsep}%
385 \fi

```

When the documentclass uses a non-zero `\parskip` setting the `\topsep` might have a negative value to compensate for that. Therefore we add `\parskip` to `\abovedisplayskip`.

```

386 \addtolength\abovedisplayskip{\parskip}%
387 \setlength\belowdisplayskip{\abovedisplayskip}%
388 \setlength\belowdisplayshortskip{\abovedisplayskip}%
389 \setlength\abovedisplayshortskip{\abovedisplayskip}%
390 $$\everycr{}\halign to\linewidth% $$
391 \bgroup
392 \hskip\@centering
393 $\displaystyle\tabskip\z@skip{##}$\@eqnsele%
394 \global\@eqcnt\@ne \hskip \tw@\arraycolsep \hfil${##}$\hfil%
395 \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
396 $\displaystyle{##}$\hfil \tabskip\@centering%
397 \global\@eqcnt\thr@@
398 \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}%
399 {\@eqnocr
400 \egroup
401 \global\advance\c@equation\m@ne$$$ $$$
402 \@ignoretrue
403 }
404 \fleqn)

```

# File A

## ltlists.dtx

### 54 List, and related environments

The generic commands for creating an indented environment – `enumerate`, `itemize`, `quote`, etc – are:

```
\list{<LABEL>}{<COMMANDS>} ... \endlist
```

which can be invoked by the user as the list environment. The LABEL argument specifies item labeling. COMMANDS contains commands for changing the horizontal and vertical spacing parameters.

Each item of the environment is begun by the command `\item[ITEMLABEL]` which produces an item labeled by ITEMLABEL. If the argument is missing, then the LABEL argument of the `\list` command is used as the item label.

The label is formed by putting `\makelabel{<ITEMLABEL>}` in an hbox whose width is either its natural width or else `\labelwidth`, whichever is larger. The `\list` command defines `\makelabel` to have the default definition:

```
\makelabel{<ARG>} == BEGIN \hfil ARG END
```

which, for a label of width less than `\labelwidth`, puts the label flushright, `\labelsep` to the left of the item's text. However, `\makelabel` can be `\let` to another command by the `\list`'s COMMANDS argument.

A `\usecounter{<foo>}` command in the second argument causes the counter `foo` to be initialized to zero, and stepped by every `\item` command without an argument. (`\label` commands within the list refer to this counter.)

When you leave a list environment, returning either to an enclosing list or normal text mode, LaTeX begins a new paragraph if and only if you leave a blank line after the `\end` command. This is accomplished by the `\@endparenv` command.

Blank lines are ignored every other reasonable place—i.e.:

- Between the `\begin{list}` and the first `\item`,
- Between the `\item` and the text of that item.
- Between the end of the last item and the `\end{list}`.

For an environment like `quotation`, in which items are not labeled, the entire environment is a single item. It is defined by letting `\quotation == \list{}{...}\item\relax`. (Note the `\relax`, there in case the first character in the environment is a '['.) The spacing parameters provide a great deal of flexibility in designing the format, including the ability to let the indentation of the first paragraph be different from that of the subsequent ones.

The `trivlist` environment is equivalent to a list environment whose second argument sets the following parameter values:

`\leftmargin = 0`: causes no indentation of left margin

`\labelwidth = 0`: see below for precise effect this has.

`\itemindent = 0`: with a null label, makes first paragraph have no indentation. Succeeding paragraphs have `\parindent` indentation. To give first paragraph same indentation, set `\itemindent = \parindent` before the `\item[]`.

Every `\item` in a trivlist environment must have an argument—in many cases, this will be the null argument (`\item[]`). The trivlist environment is mainly used for paragraphing environments, like `verbatim`, in which there is no margin change. It provides the same vertical spacing as the list environment, and works reasonably well when it occurs immediately after an `\item` command in an enclosing list.

## 54.1 List and Trivlist

The following variables are used inside a list environment:

`\totalleftmargin` The distance that the prevailing left margin is indented from the outermost left margin,

`\linewidth` The width of the current line. Must be initialized to `\hsize`.

`\listdepth` A count for holding current list nesting depth.

`\makelabel` A macro with a single argument, used to generate the label from the argument (given or implied) of the `\item` command. Initialized to `\@mklab` by the `\list` command. This command must produce some stretch—i.e., an `\hfil`.

`\@inlabel` A switch that is false except between the time an `\item` is encountered and the time that `TEX` actually enters horizontal mode. Should be tested by commands that can be messed up by the list environment's use of `\everypar`.

`\box\@labels` When `\@inlabel = true`, it holds the labels to be put out by `\everypar`.

`\@nparitem` A switch set by `\list` when `\@inlabel = true`. Handles the case of a `\list` being the first thing in an item.

`\@nparlist` A switch set true for a list that begins an item. No `\topsep` space is added before or after `\item`'s such a list.

`\@newlist` Set true by `\list`, set false by the first text (by `\everypar`).

`\@noitemarg` Set true when executing an `\item` with no explicit argument. Used to save space. To save time, make two separate `\@item` commands.

`\@nmbulist` Set true by `\usecounter` command, causes list to be numbered.

`\@listctr` `\def`'ed by `\usecounter` to name of counter.

`\@noskipsec` A switch set true by a sectioning command when it is creating an in-text heading with `\everypar`.

Throughout a list environment, `\hsize` is the width of the current line, measured from the outermost left margin to the outermost right margin. Environments like `tabbing` should use `\linewidth` instead of `\hsize`.



Here are the parameters of a list that can be set by commands in the `\list`'s COMMANDS argument. These parameters are all TeX skips or dimensions (defined by `\newskip` or `\newdimen`), so the usual TeX or L<sup>A</sup>T<sub>E</sub>X commands can be used to set them. The commands will be executed in vmode if and only if the `\list` was preceded by a `\par` (or something like an `\end{list}`), so the spacing parameters can be set according to whether the list is inside a paragraph or is its own paragraph.

## 54.2 Vertical Spacing (skips)

`\topsep`: Space between first item and preceding paragraph.

`\partopsep`: Extra space added to `\topsep` when environment starts a new paragraph (is called in vmode).

`\itemsep`: Space between successive items.

`\parsep`: Space between paragraphs within an item – the `\parskip` for this environment.

## 54.3 Penalties

`\@beginparpenalty`: put at the beginning of a list

`\@endparpenalty`: put at end of list

`\@itempenalty`: put between items.

## 54.4 Horizontal Spacing (dimens)

`\leftmargin`: space between left margin of enclosing environment (or of page if top level list) and left margin of this list. Must be nonnegative.

`\rightmargin`: analogous.

`\listparindent`: extra indentation at beginning of every paragraph of a list except the one started by the `\item` command. May be negative! Usually, labeled lists have `\listparindent` equal to zero.

`\itemindent`: extra indentation added right BEFORE an item label.

`\labelwidth`: nominal width of box that contains the label. If the natural width of the label  $\leq$  `\labelwidth`, then the label is flushed right inside a box of width `\labelwidth` (with an `\hfil`). Otherwise, a box of the natural width is employed, which causes an indentation of the text on that line.

`\labelsep`: space between end of label box and text of first item.

## 54.5 Default Values

Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, one of the commands `\@listi`, `\@listii`, ... , `\@listvi` is called, depending upon the current level of the list. The `\@list ...` commands should be defined by the document style. A convention that the document style should follow is to set `\leftmargin` to `\leftmargini`, ..., `\leftmarginvi` for the appropriate level. Items that aren't changed may be left alone, but everything that could possibly be changed must be reset.

```

\list{LABEL}{COMMANDS} ==
 BEGIN
 if \@listdepth > 5
 then LaTeX error: 'Too deeply nested'
 else \@listdepth :=G \@listdepth + 1
 fi
 \rightmargin := 0pt
 \listparindent := 0pt
 \itemindent := 0pt
 \eval{@list \romannumeral\the\@listdepth} %% Set default values:
 \@itemlabel :=L LABEL
 \makelabel == \@mklab
 @nmbrlist :=L false
 COMMANDS

 \@trivlist % commands common to \list and
\trivlist

 \parskip :=L \parsep
 \parindent :=L \listparindent
 \linewidth :=L \linewidth - \rightmargin - \leftmargin
 \@totalleftmargin :=L \@totalleftmargin + \leftmargin
 \parshape 1 \@totalleftmargin \linewidth
 \ignorespaces % gobble space up to \item
 END

\endlist == BEGIN \@listdepth :=G \@listdepth -1
 \endtrivlist
 END

\@trivlist ==
 BEGIN
 if @newlist = T then \@noitemerr fi
 %% This command removed for some forgotten
reason.
 \@topsepadd :=L \topsep
 if @noskipsec then leave vertical mode fi %% Added 11 Jun 85
 if vertical mode
 then \@topsepadd :=L \@topsepadd + \partopsep
 else \unskip \par % remove glue from end of last line

```

```

 fi
 if @inlabel = true
 then @noperitem :=L true
 @noperlist :=L true
 else @noperlist :=L false
 \@topsep :=L \@topsepadd
 fi
 \@topsep :=L \@topsep + \parskip %% Change 4 Sep 85
 \leftskip :=L 0pt % Restore paragraphing
parameters
 \rightskip :=L \@rightskip
 \parfillskip :=L 0pt + 1fil

NOTE: \@setpar called on every \list in case \par has been
temporarily munged before the \list command.
 \@setpar{if @newlist = false then {\@@par} fi}
 \@newlist :=G T
 \@outerparskip :=L \parskip
END

\trivlist ==
BEGIN
 \parsep := \parskip
 @nmbrlist := F
 \@trivlist
 \labelwidth := 0
 \leftmargin := 0
 \itemindent := \parindent
 \@itemlabel :=L "empty" %% added 93/12/13
 \makelabel{LABEL} == LABEL
END

\endtrivlist ==
BEGIN
 if @inlabel = T then \indent fi
 if horizontal mode then \unskip \par fi
 if @noperlist = true
 else if \lastskip > 0
 then \@tempskipa := \lastskip
 \vskip - \lastskip
 \vskip \@tempskipa - \@outerparskip + \parskip
 fi
 \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

\topskip
\partopsep 1 (*2kernel)
\itemsep 2 \newskip\topsep
\parsep 3 \newskip\partopsep
\@topsep 4 \newskip\itemsep
\@topsepadd 5 \newskip\parsep
\outerparskip 6 \newskip\@topsep
7 \newskip\@topsepadd
8 \newskip\@outerparskip

```

```

\leftmargin
\rightmargin 9 \newdimen\leftmargin
\listparindent 10 \newdimen\rightmargin
\itemindent 11 \newdimen\listparindent
\labelwidth 12 \newdimen\itemindent
\labelsep 13 \newdimen\labelwidth
\@totalleftmargin 14 \newdimen\labelsep
 15 \newdimen\linewidth
 16 \newdimen\@totalleftmargin \@totalleftmargin=\z@

\leftmargini
\leftmarginii 17 \newdimen\leftmargini
\leftmarginiii 18 \newdimen\leftmarginii
\leftmarginiv 19 \newdimen\leftmarginiii
\leftmarginv 20 \newdimen\leftmarginiv
\leftmarginvi 21 \newdimen\leftmarginv
 22 \newdimen\leftmarginvi

\@listdepth
\@itempenalty 23 \newcount\@listdepth \@listdepth=0
\@beginparpenalty 24 \newcount\@itempenalty
\@endparpenalty 25 \newcount\@beginparpenalty
 26 \newcount\@endparpenalty

\@labels
 27 \newbox\@labels

\if@inlabel
\@inlabelfalse 28 \newif\if@inlabel \@inlabelfalse
\@inlabeltrue
\if@newlist
\@newlistfalse 29 \newif\if@newlist \@newlistfalse
\@newlisttrue
\if@noparitem
\@noparitemfalse 30 \newif\if@noparitem \@noparitemfalse
\@noparitemtrue
\if@noparlist
\@noparlistfalse 31 \newif\if@noparlist \@noparlistfalse
\@noparlisttrue
\if@noitemarg
\@noitemargfalse 32 \newif\if@noitemarg \@noitemargfalse
\@noitemargtrue
\if@newlist
\@newlistfalse 33 \newif\if@nbrlist \@nbrlistfalse
\@newlisttrue
\list
 34 \def\list#1#2{%
 35 \ifnum \@listdepth >5\relax
 36 \@toodeep
 37 \else
 38 \global\advance\@listdepth\@ne
 39 \fi
 40 \rightmargin\z@

```

```

41 \listparindent\z@
42 \itemindent\z@
43 \csname @list\romannumeral\the\@listdepth\endcsname
44 \def\@itemlabel{#1}%
45 \let\makelabel\@mklab
46 \@nmbrrlistfalse
47 #2\relax
48 \@trivlist
49 \parskip\parsep
50 \parindent\listparindent
51 \advance\linewidth -\rightmargin
52 \advance\linewidth -\leftmargin
53 \advance\@totalleftmargin \leftmargin
54 \parshape \@ne \@totalleftmargin \linewidth
55 \ignorespaces}

```

`\par@deathcycles`

```
56 \newcount\par@deathcycles
```

`\@trivlist` Because `\par` is sometimes made a no-op it is possible for a missing `\item` to produce a loop that does not fill memory and so never gets trapped by T<sub>E</sub>X. We thus need to trap this here by setting `\par` to count the number of times a paragraph ii is called with no progress being made started.

```

57 \def\@trivlist{%
58 \if@noskipsec \leavevmode \fi
59 \@topsepadd \topsep
60 \ifvmode
61 \advance\@topsepadd \partopsep
62 \else
63 \unskip \par
64 \fi
65 \if@inlabel
66 \@nparitemtrue
67 \@nparlisttrue
68 \else
69 \if@newlist \@noitemerr \fi
70 \@nparlistfalse
71 \@topsep \@topsepadd
72 \fi
73 \advance\@topsep \parskip
74 \leftskip \z@skip
75 \rightskip \@rightskip
76 \parfillskip \@flushglue
77 \par@deathcycles \z@
78 \setpar{\if@newlist
79 \advance\par@deathcycles \@ne
80 \ifnum \par@deathcycles >\@m
81 \@noitemerr
82 {\@@par}%
83 \fi
84 \else
85 {\@@par}%
86 \fi}%
87 \global \@newlisttrue

```

```
88 \@outerparskip \parskip}
```

```
\trivlist
```

```
89 \def\trivlist{%
90 \parsep\parskip
91 \@nmbrrlistfalse
92 \@trivlist
93 \labelwidth\z@
94 \leftmargin\z@
95 \itemindent\z@
```

We initialise \@itemlabel so that a trivlist with an \item not having an optional argument doesn't produce an error message.

```
96 \let\@itemlabel\@empty
97 \def\makelabel##1{##1}}
```

```
\endlist
```

```
98 \def\endlist{%
99 \global\advance\@listdepth\m@ne
100 \endtrivlist}
```

The definition of \trivlist used to be in ltspace.dtx so that other commands could be 'let to it'. They now use \def.

```
\endtrivlist
```

```
101 \def\endtrivlist{%
102 \if@inlabel
103 \leavevmode
104 \global \@inlabelfalse
105 \fi
106 \if@newlist
107 \@noitemerr
108 \global \@newlistfalse
109 \fi
110 \ifhmode\unskip \par
```

We also check if we are in math mode and issue an error message if so (hoping that \@currenvir resolves suitably). Otherwise the usual "perhaps a missing item" error will get triggered later which is confusing.

```
111 \else
112 \@inmatherr{\end{\@currenvir}}}%
113 \fi
114 \if@nparlist \else
115 \ifdim\lastskip >\z@
116 \@tempskipa\lastskip \vskip -\lastskip
117 \advance\@tempskipa\parskip \advance\@tempskipa -\@outerparskip
118 \vskip\@tempskipa
119 \fi
120 \@endparenv
121 \fi
122 }
```

```
\@endparenv To suppress the paragraph indentation in text immediately following a paragraph-
\@doendpe making environment, \everypar is changed to remove the space, and \par is
```



redefined to restore `\everypar`. Instead of redefining `\par` and `\everypar`, `\@endparenv` was changed to set the `@endpe` switch, letting `\end` redefine `\par` and `\everypar`.

This allows paragraph-making environments to work right when called by other environments. (Changed 27 Oct 86)

```
123 \def\@endparenv{%
124 \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}

125 (latexrelease)\IncludeInRelease{2015/01/01}{\@doendpe}{clubpenalty fix}%
126 \def\@doendpe{\@endpetrue
127 \def\par{\@restorepar
```

If a section heading changes `\clubpenalty` to keep lines after it together then this modification is restored via the `\everypar` mechanism at the start of the next paragraph. As we destroy the contents of this token here we explicitly set `\clubpenalty` back to its default.

```
128 \clubpenalty\@clubpenalty
129 \everypar{\par\@endpefalse}\everypar
```

Use `\setbox0=\lastbox` instead of `\hskip -\parindent` so that a `\noindent` becomes a no-op when used before a line immediately following a list environment(23 Oct 86).

```
130 {\setbox\z@\lastbox}%
131 \everypar{\@endpefalse}}
132 (latexrelease)\EndIncludeInRelease

133 (latexrelease)\IncludeInRelease{0000/00/00}{\@doendpe}{clubpenalty fix}%
134 (latexrelease)\def\@doendpe{\@endpetrue
135 (latexrelease) \def\par{\@restorepar\everypar{\par\@endpefalse}\everypar
136 (latexrelease) {\setbox\z@\lastbox}\everypar{\@endpefalse}}
137 (latexrelease)\EndIncludeInRelease
```

```
\if@endpe
\@endpefalse 138 \newif\if@endpe
\@endpeltrue 139 \endpefalse
```

```
\@mklab
140 \def\@mklab#1{\hfil #1}
```

```
\item
141 \def\item{%
142 \@inmatherr\item
143 \@ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

```
\@donoparitem
144 \def\@donoparitem{%
145 \@noparitemfalse
146 \global\setbox\@labels\hbox{\hskip -\leftmargin
147 \unhbox\@labels
148 \hskip \leftmargin}%
149 \if@minipage\else
150 \@tempskipa\lastskip
151 \vskip -\lastskip
```

```

152 \advance\@tempskipa\@outerparskip
153 \advance\@tempskipa -\parskip
154 \vskip\@tempskipa
155 \fi}

```

`\@item`

```

156 \def\@item[#1]{%
157 \if@noperitem
158 \@donoperitem
159 \else
160 \if@inlabel
161 \indent \par
162 \fi
163 \ifhmode
164 \unskip\unskip \par
165 \fi
166 \if@newlist
167 \if@nobreak
168 \@nbitem
169 \else
170 \addpenalty\@beginparpenalty
171 \addvspace\@topsep
172 \addvspace{-\parskip}%
173 \fi
174 \else
175 \addpenalty\@itempenalty
176 \addvspace\itemsep
177 \fi
178 \global\@inlabeltrue
179 \fi
180 \everypar{%
181 \@minipagefalse
182 \global\@newlistfalse

```

This `\if@inlabel` check is needed in case an item starts of inside a group so that `\everypar` does not become empty outside that group. `nobreakfalse`, etc etc.

```

183 \if@inlabel
184 \global\@inlabelfalse

```

The paragraph indent is now removed by using `\setbox...` since this makes `\noindent` a no-op here, as it should be. Thus the following comment is redundant but is left here for the sake of future historians: this next command was changed from an `hskip` to a `kern` to avoid a break point after the `parindent` box: the `skip` could cause a line-break if a very long label occurs in `raggedright` setting.

If `\noindent` was used after `\item` want to cancel the `\itemindent` skip. This case can be detected as the indentation box will be void.

```

185 {\setbox\z@\lastbox
186 \ifvoid\z@
187 \kern-\itemindent
188 \fi}%
189 \box\@labels
190 \penalty\z@
191 \fi

```

This code is intended to prevent a page break after the first line of an item that comes immediately after a section title. It may be sensible to always forbid a page break after one line of an item? As with all such settings of `\clubpenalty` it is local so will have no effect if the item starts in a group.

Only resetting `\@nobeak` when it is true is now essential since now it is sometimes set locally.

```

192 \if@nobeak
193 \@nobeakfalse
194 \clubpenalty \@M
195 \else
196 \clubpenalty \@clubpenalty
197 \everypar{}%
198 \fi}%

199 \if@noitemarg
200 \@noitemargfalse
201 \if@nmbrrlist

202 \refstepcounter\@listctr
203 \fi
204 \fi

```

We use `\sbox` to support colour commands.

```

205 \sbox\@tempboxa{\makelabel{#1}}%
206 \global\setbox\@labels\hbox{%
207 \unhbox\@labels
208 \hskip \itemindent
209 \hskip -\labelwidth
210 \hskip -\labelsep
211 \ifdim \wd\@tempboxa >\labelwidth
212 \box\@tempboxa

213 \else
214 \hbox to\labelwidth {\unhbox\@tempboxa}%
215 \fi
216 \hskip \labelsep}%
217 \ignorespaces}

```

`\makelabel`

```

218 \def\makelabel#1{%
219 \@latex@error{Lonely \string\item--perhaps a missing
220 list environment}\@ehc}

```

`\@nbitem`

```

221 \def\@nbitem{%
222 \@tempskipa\@outerparskip
223 \advance\@tempskipa -\parskip
224 \addvspace\@tempskipa}

```

`\usecounter`

```

225 \def\usecounter#1{\@nmbrrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}

```

## 54.6 Itemize and Enumerate

Enumeration is done with four counters: `enumi`, `enumii`, `enumiii` and `enumiv`, where `enumN` controls the numbering of the Nth level enumeration. The label is generated by the commands `\labelenumi` ... `\labelenumiv`, which should be defined by the document style. Note that `\p@enumN\theenumN` defines the output of a `\ref` command. A typical definition might be:

```
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}
\def\labelenumii{(\theenumii)}
```

which will print the labels as ‘(a)’, ‘(b)’, ... and print a `\ref` as ‘3a’.

The item numbers are moved to the right of the label box, so they are always a distance of `\labelsep` from the item.

`\@enumdepth` holds the current enumeration nesting depth.

Itemization is controlled by four commands: `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`. To cause the second-level list to be bulleted, you just define `\labelitemii` to be `\bullet`. `\@itemspacing` and `\@itemdepth` are the analogs of `\@enumspacing` and `\@enumdepth`.

```
\enumerate ==
BEGIN
 if \@enumdepth > 3
 then errormessage: "Too deeply nested".
 else \@enumdepth :=L \@enumdepth + 1
 \@enumctr :=L eval(enum@\romannumeral\the\@enumdepth)
 \list{\label{\@enumctr}}
 {\usecounter{\@enumctr}
 \makelabel{LABEL} == \hss \llap{LABEL}}
 fi
 END

\endenumerate == \endlist
```

```
\@enumdepth
226 \newcount\@enumdepth \@enumdepth = 0

\c@enumi
\c@enumii 227 \@definecounter{enumi}
\c@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

\itemize ==
BEGIN
 if \@itemdepth > 3
 then errormessage: 'Too deeply nested'.
 else \@itemdepth :=L \@itemdepth + 1
 \@itemitem ==
eval(labelitem\romannumeral\the\@itemdepth)
 \list{\@nameuse{\@itemitem}}
 {\makelabel{LABEL} == \hss \llap{LABEL}}
 fi
 END

\enditemize == \endlist

\@itemdepth
241 \newcount\@itemdepth \@itemdepth = 0

itemize
242 \def\itemize{%
243 \ifnum \@itemdepth >\thr@@\toodeep\else
244 \advance\@itemdepth\@ne
245 \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%

246 \expandafter
247 \list
248 \csname\@itemitem\endcsname
249 {\def\makelabel##1{\hss\llap{##1}}}%
250 \fi}

251 \let\enditemize =\endlist
252 \endkernel)

```

# File B

## ltboxes.dtx

### 55 L<sup>A</sup>T<sub>E</sub>X Box commands

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\makebox</code>    | <code>\makebox[⟨wid⟩][⟨pos⟩]{⟨obj⟩}</code><br>Puts <code>⟨obj⟩</code> in an <code>\hbox</code> of width <code>⟨wid⟩</code> , positioned by <code>⟨pos⟩</code> .<br>The possible <code>⟨pos⟩</code> are:<br>s stretched,<br>l flushleft,<br>r flushright,<br>c (default) centred.<br>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.<br><code>\makebox(⟨x⟩,⟨y⟩)[⟨pos⟩]{⟨obj⟩}</code><br>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 <sub>E</sub> X box register.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\savebox</code>    | <code>\savebox{⟨cmd⟩ ...</code> : <code>⟨cmd⟩</code> is defined to be a T <sub>E</sub> X box register, and the '...' are any <code>\makebox</code> arguments. It is like <code>\makebox</code> , except it doesn't produce text but saves the value in <code>\box ⟨cmd⟩</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>\sbox</code>       | <code>\sbox{⟨cmd⟩}{⟨obj⟩}</code> is an efficient abbreviation for <code>\savebox{⟨cmd⟩}{⟨obj⟩}</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>lrbox</code>       | <code>\begin{lrbox}{⟨cmd⟩}⟨text⟩\end{lrbox}</code> is equivalent to <code>\sbox{⟨cmd⟩}{⟨text⟩}</code> except that any white space at the beginning and end of <code>⟨text⟩</code> is ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>\framebox</code>   | <code>\framebox ...</code> : like <code>\makebox</code> , except it puts a 'frame' around the box. The frame is made of lines of thickness <code>\fboxrule</code> , separated by space <code>\fboxsep</code> from the text – except for <code>\framebox(X,Y) ...</code> , where the thickness of the lines is as for the picture environment, and there is no separation added.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <code>\fbox</code>       | <code>\fbox{⟨obj⟩}</code> is an abbreviation for <code>\framebox{⟨obj⟩}</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>\parbox</code>     | <code>\parbox[⟨pos⟩][⟨height⟩][⟨inner-pos⟩]{⟨width⟩}{⟨text⟩}</code> : Makes a box with <code>\hsize ⟨width⟩</code> , positioned by <code>⟨pos⟩</code> as follows: c : <code>\vcenter</code> (placed in <code>\$. . . \$</code> if not in math mode) b : <code>\vbox</code> t : <code>\vtop</code> default value is c. Sets <code>\hsize := ⟨width⟩</code> and calls <code>\@parboxrestore</code> , which does the following: Restores the original definitions of:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

```

\par
\\
\~
\'
\‘
\=
Resets the following parameters:
\parindent = 0pt
\parskip = 0pt
\linewidth = \hsize
\@totalleftmargin = 0pt
\leftskip = 0pt
\rightskip = 0pt
\@rightskip = 0pt
\parfillskip = 0pt plus 1fil
\lineskip = \normallineskip
\baselineskip = \normalbaselineskip
Calls \sloppy
Note: \@arrayparboxrestore same as \@parboxrestore but it doesn't re-
store \\.
minipage minipage : Similar to \parbox, except it also makes this look like a page by
setting
\textwidth == \columnwidth == box width
changes footnotes by redefining:
\@mpfn == mpfootnote
\thempfn == \thempfootnote
\@footnotetext == \@mpfootnotetext
resets the following list environment parameters:
\@listdepth == \@mplistdepth
where \@mplistdepth is initialized to zero,
and executes \@minipagerestore to allow the document style to reset any
other parameters it desires. It sets @minipage true, and resets \everypar to set it
false. This switch keeps \addvspace from putting space at the top of a minipage.
Change added 24 May 89: \minipage sets @minipage globally; \endminipage
resets it false.
\rule \rule[\langle raised \rangle]{\langle width \rangle}{\langle height \rangle} : Makes a \langle width \rangle * \langle height \rangle rule, raised
\langle raised \rangle.
\underline \underline{\langle text \rangle} : Makes an underlined hbox with \langle text \rangle in it.
\raisebox \raisebox{\langle distance \rangle}[\langle height \rangle][\langle depth \rangle]{\langle box \rangle} :
Raises \langle box \rangle up by \langle distance \rangle length (down if \langle distance \rangle negative). Makes TEX
think that the new box extends \langle height \rangle above the line and \langle depth \rangle below, for a
total vertical length of \langle height \rangle + \langle depth \rangle. Default values of \langle height \rangle & \langle depth \rangle =
actual height and depth of box in new position.

1 (*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>\EndIncludeInRelease
22 <*2ekernel)

```

`\mbox` The basic horizontal box command for L<sup>A</sup>T<sub>E</sub>X.

```
23 \long\def\mbox#1{\leavevmode\hbox{#1}}
```

`\@makebox` Look for a possible second optional argument (defaults to c).

```
24 \def\@makebox[#1]{%
25 \@ifnextchar [#{@imakebox[#1]}{\@imakebox[#1][c]}}
```

`\@begin@tempboxa` Helper macro for supporting `\height`, `\width` etc. Grab #1 into `\@tempboxa` and measure it.

```

26 \long\def\@begin@tempboxa#1#2{%
27 \begingroup
28 \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
29 \def\width{\wd\@tempboxa}%
30 \def\height{\ht\@tempboxa}%
31 \def\depth{\dp\@tempboxa}%
32 \let\totalheight\@ovri
33 \totalheight\height
34 \advance\totalheight\depth}

```

`\@end@tempboxa` End the group started by `\@begin@tempboxa`, so that the scope of `\height` only includes the ‘length’ argument to the user-command.

```
35 \let\@end@tempboxa\endgroup
```

`\bm@c` Set up spacing.

```

\bm@l 36 \def\bm@c{\hss\unhbox\@tempboxa\hss}
\bm@r 37 \def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
\bm@s 38 \def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
\bm@t 39 \def\bm@s{\unhbox\@tempboxa}

```

`\bm@b`

`\@imakebox` Internal form of `\makebox`.

```

40 \long\def\@imakebox[#1][#2]#3{%
41 \@begin@tempboxa\hbox{#3}%
42 \setlength\@tempdima{#1}% support calc
43 \hb@xt@\@tempdima{\csname bm@#2\endcsname}%
44 \@end@tempboxa}

```



`\@makepicbox` Picture mode form of `\makebox`.  
45 `\def\@makepicbox(#1,#2){%`  
46 `\@ifnextchar[{\@imakepicbox(#1,#2)}{\@imakepicbox(#1,#2) []}]}`

`\imakepicbox` picture mode version  
47 `\long\def\imakepicbox(#1,#2)[#3]#4{%`  
48 `\vbox to#2\unitlength`  
49 `{\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss`  
50 `\let\mb@t\vss`  
51 `\@tfor\reserved@a :=#3\do{%`  
52 `\if s\reserved@a`  
53 `\let\mb@l\relax\let\mb@r\relax`  
54 `\else`  
55 `\expandafter\let\csname mb@\reserved@a\endcsname\relax`  
56 `\fi}%`  
57 `\mb@t`  
58 `\hb@xt@ #1\unitlength{\mb@l #4\mb@r}%`  
59 `\mb@b`

This kern ensures that a `b` option aligns on the bottom of the text rather than the baseline. this is the documented behaviour in the `LATEXBook`. The kern is removed in compatibility mode.

60 `\kern\z@}`

`\set@color` This macro is initially a no-op, but the colour package will redefine it to insert a `\special`.

61 `\let\set@color\relax`

`\color@begingroup` These macros are initially a no-op, but the colour package will redefine them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`,  
`\color@endgroup` `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and (*set to main document colour*) respectively.

`\color@setgroup` 62 `\let\color@begingroup\relax`  
`\normalcolor` 63 `\let\color@endgroup\relax`  
`\color@hbox` 64 `\let\color@setgroup\relax`  
`\color@vbox` 65 `\let\normalcolor\relax`  
`\color@endbox` 66 `\let\color@hbox\relax`  
67 `\let\color@vbox\relax`  
68 `\let\color@endbox\relax`

`\newsavebox` Allocate a new ‘savebox’.

69 `\def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}`

`\savebox` Save `#1` in a box register.

70 `\</2kernel`  
71 `\(latexrelease)\IncludeInRelease{2015/01/01}%`  
72 `\(latexrelease) \savebox}{Make \savebox robust}%`  
73 `\*2kernel | latexrelease`  
74 `\DeclareRobustCommand\savebox[1]{%`  
75 `\@ifnextchar(%)`  
76 `{\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%`  
77 `\</2kernel | latexrelease`  
78 `\(latexrelease)\EndIncludeInRelease`

```

79 (latexrelease)\IncludeInRelease{0000/00/00}%
80 (latexrelease) {\savebox}{Make \savebox robust}%
81 (latexrelease)\def\savebox#1{%
82 (latexrelease) \ifnextchar(%)
83 (latexrelease) {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
84 (latexrelease)\EndIncludeInRelease
85 (*2ekernel)

\sbox Save #1 in a box register.
86 \long\def\sbox#1#2{\setbox#1\hbox{%
87 \color@setgroup#2\color@endgroup}}

\@savebox Look for second optional argument.
88 \def\@savebox#1[#2]{%
89 \@ifnextchar [{\@isavebox#1[#2]} {\@isavebox#1[#2] [c]}}

\@isavebox
90 \long\def\@isavebox#1[#2] [#3] #4{%
91 \sbox#1{\@imakebox[#2] [#3]{#4}}

\@savepicbox Picture mode version of \savebox.
92 \def\@savepicbox#1(#2,#3){%
93 \@ifnextchar [%]
94 {\@isavepicbox#1(#2,#3)} {\@isavepicbox#1(#2,#3) []}}

\@isavepicbox Picture mode version of \savebox.
95 \long\def\@isavepicbox#1(#2,#3) [#4] #5{%
96 \sbox#1{\@imakepicbox(#2,#3) [#4]{#5}}}

\lrbox lrbox: the new environment form of \sbox. Use \aftergroup tricks to enable a
local assignment to be made to the box, in a way that it still has an effect outside
the lrbox environment.
97 \def\lrbox#1{%
98 \edef\reserved@a{%
99 \endgroup
100 \setbox#1\hbox{%
101 \begingroup\aftergroup}%
102 \def\noexpand\@currenvir{\@currenvir}%
103 \def\noexpand\@currencline{\on@line}}%
104 \reserved@a
105 \@endpefalse
106 \color@setgroup
107 \ignorespaces}

\endlrbox End the lrbox environment.
108 \def\endlrbox{\unskip\color@endgroup}

\usebox unchanged
109 \def\usebox#1{\leavevmode\copy #1\relax}

\frame The following definition of \frame was written by Pavel Curtis (Extra space
removed 14 Jan 88) RmS 92/08/24: Replaced occurrence of \@halfwidth by
\@wholewidth

```

```

110 \long\def\frame#1{%
111 \leavevmode
112 \hbox{%
113 \hskip-\@wholewidth
114 \vbox{%
115 \vskip-\@wholewidth
116 \hrule \@height\@wholewidth
117 \hbox{%
118 \vrule\@width\@wholewidth
119 #1%
120 \vrule\@width\@wholewidth}%
121 \hrule\@height\@wholewidth
122 \vskip-\@wholewidth}%
123 \hskip-\@wholewidth}}

\fbxrule user level parameters,
\fbxsep 124 \newdimen\fbxrule
125 \newdimen\fbxsep

\fbx Abbreviated framed box command.
126 \long\def\fbx#1{%
127 \leavevmode
128 \setbox\@tempboxa\hbox{%
129 \color@begingroup
130 \kern\fbxsep{#1}\kern\fbxsep
131 \color@endgroup}%
132 \@frameb@x\relax}

\framebox Framed version of \makebox.
133 </2ekernel>
134 <latexrelease>\IncludeInRelease{2015/01/01}%
135 <latexrelease> {\framebox}{Make \framebox robust}%
136 <*2ekernel | latexrelease>
137 \DeclareRobustCommand\framebox{%
138 \@ifnextchar(%)
139 \@framepicbox{\@ifnextchar[\@framebox\fbx]}%
140 </2ekernel | latexrelease>
141 <latexrelease>\EndIncludeInRelease
142 <latexrelease>\IncludeInRelease{0000/00/00}%
143 <latexrelease> {\framebox}{Make \framebox robust}%
144 <latexrelease>\def\framebox{%
145 <latexrelease> \@ifnextchar(%)
146 <latexrelease> \@framepicbox{\@ifnextchar[\@framebox\fbx]}%
147 <latexrelease>\EndIncludeInRelease
148 <*2ekernel>

\@framebox Deal with optional arguments.
149 \def\@framebox[#1]{%
150 \@ifnextchar[%]
151 {\@framebox[#1]}%
152 {\@framebox[#1][c]}}

\@ifframebox The handling the optional arguments. In order to set the whole box, including
the frame to the specified dimension, we first determine that dimension from the
natural size of the text, #3. calculated width.

```

```

153 \long\def\@iframebox[#1][#2]#3{%
154 \leavevmode
155 \@begin@tempboxa\hbox{#3}%
156 \setlength\@tempdima{#1}%
157 \setbox\@tempboxa\hb@xt@\@tempdima
158 {\kern\fboxsep\csname bm@#2\endcsname\kern\fboxsep}%
159 \@frameb@x{\kern-\fboxrule}%
160 \@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.

```

161 \def\@frameb@x#1{%
162 \@tempdima\fboxrule
163 \advance\@tempdima\fboxsep
164 \advance\@tempdima\dp\@tempboxa
165 \hbox{%
166 \lower\@tempdima\hbox{%
167 \vbox{%
168 \hrule\@height\fboxrule
169 \hbox{%
170 \vrule\@width\fboxrule
171 #1%
172 \vbox{%
173 \vskip\fboxsep
174 \box\@tempboxa
175 \vskip\fboxsep}%
176 #1%
177 \vrule\@width\fboxrule}%
178 \hrule\@height\fboxrule}%
179 }%
180 }%
181 }

```

`\@framepicbox` Picture mode version.

```

182 \def\@framepicbox(#1,#2){%
183 \@ifnextchar[{\@iframepicbox(#1,#2)}{\@framepicbox(#1,#2) []}]

```

`\@iframepicbox` Picture mode version.

```

184 \long\def\@iframepicbox(#1,#2)[#3]#4{%
185 \frame{\@imakepicbox(#1,#2)[#3]{#4}}

```

`\parbox` The main vertical-box command for L<sup>A</sup>T<sub>E</sub>X.

```

186 </2kernel>
187 <latexrelease>\IncludeInRelease{2015/01/01}%
188 <latexrelease> {\parbox}{Make \parbox robust}%
189 <*2kernel | latexrelease>
190 \DeclareRobustCommand\parbox{%
191 \@ifnextchar [%]
192 \@iparbox
193 {\@iiiparbox c\relax[s]}}%
194 </2kernel | latexrelease>
195 <latexrelease>\EndIncludeInRelease
196 <latexrelease>\IncludeInRelease{0000/00/00}%
197 <latexrelease> {\parbox}{Make \parbox robust}%

```

```

198 \latexrelease\def\parbox{%
199 \latexrelease) \@ifnextchar[%]
200 \latexrelease) \@iparbox
201 \latexrelease) {\@iiparbox c\relax[s]}}%
202 \latexrelease)\EndIncludeInRelease
203 (*2kernel)

```

`\@iparbox` Optional argument handling.

```

204 \def\@iparbox[#1]{%
205 \@ifnextchar[%]
206 {\@iiparbox{#1}}%
207 {\@iiparbox{#1}\relax[s]}}

```

`\@iiparbox` Optional argument handling.

```

208 \def\@iiparbox#1[#2]{%
209 \@ifnextchar[%]
210 {\@iiparbox{#1}{#2}}%
211 {\@iiparbox{#1}{#2}[#1]}}

```

`\@iiparbox` The internal version of `\parbox`.

```

\@parboxto 212 \let\@parboxto\@empty
213 \long\def\@iiparbox#1#2[#3]#4#5{%
214 \leavevmode
215 \@pboxswfalse
216 \setlength\@tempdima{#4}%
217 \@begin@tempboxa\vbox{\hsize\@tempdima\@parboxrestore#5\@par}%
218 \ifx\relax#2\else
219 \setlength\@tempdimb{#2}%
220 \edef\@parboxto{to\the\@tempdimb}%
221 \fi
222 \if#1b\vbox
223 \else\if #1t\vtop
224 \else\ifmode\vcenter
225 \else\@pboxswtrue $\vcenter
226 \fi\fi\fi
227 \@parboxto{\let\hss\vss\let\unhbox\unvbox
228 \csname bm@#3\endcsname}%
229 \if@pboxsw \m@th$\fi
230 \@end@tempboxa}

```

`\@arrayparboxrestore` Restore various paragraph parameters.

The rationale for allowing two normally global flags to be set locally here was stated originally by Donald Arsenau and extended by Chris Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within boxes or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in `\set@nobreak`; otherwise this command will be redundant.

```

231 \def\@arrayparboxrestore{%
232 \let\if@nobreak\iffalse
233 \let\if@noskipsec\iffalse
234 \let\par\@par
235 \let\-\@dischyph

```

Redefined accents to allow changes in font encoding

```
236 \let\'@acci\let\'@accii\let\=@acciii
237 \parindent\z@ \parskip\z@skip
238 \everypar{}%
239 \linewidth\hsize
240 \@totalleftmargin\z@
241 \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
242 \parfillskip\@flushglue \lineskip\normallineskip
243 \baselineskip\normalbaselineskip
244 \sloppy}
```

`\parboxrestore` Restore various paragraph parameters, and also `\.`  
245 `\def\@parboxrestore{\arrayparboxrestore\let\\\@normalcr}`

`\if@minipage` Switch that is true at the start of a minipage.  
246 `\def\@minipagefalse{\global\let\if@minipage\iffalse}`  
247 `\def\@minipagetrue {\global\let\if@minipage\iftrue}`  
248 `\@minipagefalse`

`\minipage` Essentially an environment form of `\parbox`.

```
249 \def\minipage{%
250 \@ifnextchar[%]
251 \@iminipage
252 {\@iiiminipage c\relax[s]}}
```

`\@iminipage` Optional argument handling.

```
253 \def\@iminipage[#1]{%
254 \@ifnextchar[%]
255 {\@iiiminipage{#1}}%
256 {\@iiiminipage{#1}\relax[s]}}
```

`\@iiiminipage` Optional argument handling.

```
257 \def\@iiiminipage#1[#2]{%
258 \@ifnextchar[%]
259 {\@iiiminipage{#1}{#2}}%
260 {\@iiiminipage{#1}{#2}[#1]}}
```

`\@iiiminipage` Internal form of minipage.

```
261 \def\@iiiminipage#1#2[#3]#4{%
262 \leavevmode
263 \@pboxswfalse
264 \setlength\@tempdima{#4}%
265 \def\@mpargs{#1}{#2}[#3]{#4}%
266 \setbox\@tempboxa\vbox\bgroup
267 \color@begingroup
268 \hsize\@tempdima
269 \textwidth\hsize \columnwidth\hsize
270 \parboxrestore
271 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
272 \let\@footnotetext\mpfootnotetext
273 \let\@listdepth\@mplistdepth \@mplistdepth\z@
274 \minipagerestore
275 \setminipage}
```

```

\@minipagerestore Hook so that other styles can reset other commands in a minipage.
276 \let\@minipagerestore=\relax

\endminipage
277 \def\endminipage{%
278 \par
279 \unskip
280 \ifvoid\@mpfootins\else
281 \vskip\skip\@mpfootins
282 \normalcolor
283 \footnoterule
284 \unvbox\@mpfootins
285 \fi
286 \@minipagefalse %% added 24 May 89
287 \color@endgroup
288 \egroup
289 \expandafter\@iiparbox\@mpargs{\unvbox\@tempboxa}}

\@mplistdepth Versions of \@listdepth and \footins local to minipage.
\@mpfootins 290 \newcount\@mplistdepth
291 \newinsert\@mpfootins

\@mpfootnotetext Minipage version of \@footnotetext.
Final \strut added 27 Mar 89, on suggestion by Don Hosek
292 \long\def\@mpfootnotetext#1{%
293 \global\setbox\@mpfootins\vbox{%
294 \unvbox\@mpfootins
295 \reset@font\footnotesize
296 \hsize\columnwidth
297 \@parboxrestore
298 \protected@edef\@currentlabel
299 {\csname p@mpfootnote\endcsname\@thefnmark}%
300 \color@begingroup
301 \@makefntext{%
302 \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
303 \color@endgroup}}

304 \newif\if@pboxsw

\@rule Draw a rule of the specified size.
305 (/2ekernel)
306 (latexrelease)\IncludeInRelease{2015/01/01}%
307 (latexrelease) {\rule}{Make \rule robust}%
308 (*2ekernel | latexrelease)
309 \DeclareRobustCommand\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
310 (/2ekernel | latexrelease)
311 (latexrelease)\EndIncludeInRelease
312 (latexrelease)\IncludeInRelease{0000/00/00}%
313 (latexrelease) {\rule}{Make \rule robust}%
314 (latexrelease)\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
315 (latexrelease)\EndIncludeInRelease
316 (*2ekernel)

```

```

\@rule Internal form of \rule.
317 \def\@rule[#1]#2#3{%
318 \leavevmode
319 \hbox{%
320 \setlength\@tempdima{#1}%
321 \setlength\@tempdimb{#2}%
322 \setlength\@tempdimc{#3}%
323 \advance\@tempdimc\@tempdima
324 \vrule\@width\@tempdimb\@height\@tempdimc\@depth-\@tempdima}}

\@@underline Saved primitive \underline.
325 \let\@@underline\underline

\underline LATEX version works outside math.
326 \def\underline#1{%
327 \relax
328 \ifmode\@@underline{#1}%
329 \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}

\raisebox Raise a box, and change its vertical dimensions.
330 \</2kernel>
331 \<latexrelease>\IncludeInRelease{2015/01/01}%
332 \<latexrelease> \<raisebox>{\Make \raisebox robust}%
333 \<*/2kernel | latexrelease>
334 \DeclareRobustCommand\raisebox[1]{%
335 \leavevmode
336 \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1} []}}
337 \</2kernel | latexrelease>
338 \<latexrelease>\EndIncludeInRelease
339 \<latexrelease>\IncludeInRelease{0000/00/00}%
340 \<latexrelease> \<raisebox>{\Make \raisebox robust}%
341 \<latexrelease>\def\raisebox#1{%
342 \<latexrelease> \leavevmode
343 \<latexrelease> \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1} []}}
344 \<latexrelease>\EndIncludeInRelease
345 \<*/2kernel>

\@rsbox Optional argument handling.
346 \def\@rsbox#1[#2]{%
347 \@ifnextchar[{\@irsbox{#1}[#2]}{\@irsbox{#1}[#2]}}

\@argsbox ...

\@irsbox Internal version of \raisebox (less than two optional args).
348 \long\def\@irsbox#1[#2]#3{%
349 \@begin@tempboxa\hbox{#3}%
350 \setlength\@tempdima{#1}%
351 \ifx\#2\\\else\setlength\@tempdimb{#2}\fi
352 \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
353 \ifx\#2\\\else\ht\@tempboxa\@tempdimb\fi
354 \box\@tempboxa
355 \@end@tempboxa}

```



`\@iirsbox` Internal version of `\raisebox` (two optional args).

```

356 \long\def\@iirsbox#1[#2][#3]#4{%
357 \begin@tempboxa\hbox{#4}%
358 \setlength\@tempdima{#1}%
359 \setlength\@tempdimb{#2}%
360 \setlength\dimen@{#3}%
361 \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
362 \ht\@tempboxa\@tempdimb
363 \dp\@tempboxa\dimen@
364 \box\@tempboxa
365 \@end@tempboxa}

```

`\@finalstrut` This macro adds a special strut the *depth* of the box given as #1, and height and width 0pt. It is used for ensuring that the last line of a paragraph has the correct depth in ‘p’ columns of tables and in footnotes. In vertical mode nothing is done, as adding the strut (as done in 2.09) would start a new paragraph. It would be possible to inspect `\prevdepth` to check the depth of the just-completed paragraph, but we do not do that here. Actually we do even less now, skip the vmode test as it broke tabular ‘p’ columns. .

The `\nobreak` was added (1995/10/31) to allow hyphenation of the final word of the paragraph.

```

366 \def\@finalstrut#1{%
367 \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}

```

## 55.1 Some low-level constructs

The following commands are basically inherited from plain T<sub>E</sub>X.

`\leftline` These macros place text on a full line either centred or left or right adjusted.

```

\rightline 368 \def\@@oline{\hb@xt@\hsize}
\centerline 369 \def\leftline#1{\@@oline{#1\hss}}
\@@oline 370 \def\rightline#1{\@@oline{\hss#1}}
371 \def\centerline#1{\@@oline{\hss#1\hss}}

```

`\rlap` These macros place text to the left or right of the current reference point without taking up space.

```

372 \def\rlap#1{\hb@xt@\z@{#1\hss}}
373 \def\llap#1{\hb@xt@\z@{\hss#1}}
374 \</2kernel)

```

## File C

# lftab.dtx

## 56 Tabbing, Tabular and Array Environments

This section deals with ‘Lining It Up in Columns’. First the `tabbing` environment is defined, and then in second part, `tabular` together with its variants, `tabular*` and `array`.

Note that the `tabular` defined here is essentially the original L<sup>A</sup>T<sub>E</sub>X 2.09 version, not the extended version described in *The L<sup>A</sup>T<sub>E</sub>X Companion*. Use the `array` package to obtain the extended version.

### 56.1 tabbing

`\dimen{\@firsttab + i}` = distance of tab stop `i` from left margin  
`0 <= i <= 15` (?).

`\dimen\@firsttab` is initialized to `\@totalleftmargin`, so it starts at the prevailing left margin.

`\@maxtab` = number of highest defined tab register  
probably = `\@firsttab + 12`

`\@nxttabmar` = tab stop number of next line’s left margin

`\@curtabmar` = tab stop number of current line’s left margin

`\@curtab` = number of the current tab. At start of line,  
it equals `\@curtabmar`

`\@hightab` = largest tab number currently defined.

`\@tabpush` = depth of `\pushtab`’s

`\box\@curline` = contents of current line, excluding left margin  
skip, and excluding contents of current field

`\box\@curfield` = contents of current field

`@rjfield` = switch: T iff the last field of the line should  
be right-justified at the right margin.

`\tabbingsep` = distance left by the `\` command between the  
current position and the field that is  
“left-shifted”.

#### UTILITY MACROS

`\@stopfield` : closes the current field

`\@addfield` : adds the current field to the current line.

`\@contfield` : continues the current field

`\@startfield` : begins the next field

`\@stopline` : closes the current line and outputs it

`\@startline` : starts the next line

`\@ifatmargin` : an `\if` that is true iff the current line.

has width zero

```
\@startline ==
BEGIN
 \@curtabmar :=G \@nxttabmar
 \@curtab :=G \@curtabmar
 \box\@curline :=G null
 \@startfield
 \strut
END

\@stopline ==
BEGIN
 \unskip
 \@stopfield
 if @rjfield = T
 then @rjfield :=G F
 \@tempdima := \@totalleftmargin + \linewidth
 \hb@xt@ \@tempdima{\@itemfudge
 \hskip \dimen\@curtabmar
 \box\@curline
 \hfil
 \box\@curfield}
 else \@addfield
 \hbox {\@itemfudge
 \hskip \dimen\@curtabmar
 \box\@curline}
 fi
 END

\@startfield ==
BEGIN
 \box\@curfield :=G \hbox {
 END

\@stopfield ==
BEGIN
 }
 END

\@contfield ==
BEGIN
 \box\@curfield :=G \hbox { \unhbox\@currfield %%} brace
matching
 END
\@addfield ==
BEGIN
 \box\@curline :=G \unbox\@curline * \unbox\@curfield
 END
```

```

\@ifatmargin ==
BEGIN
 if dim of box\@curline = 0pt then
 END

\@tabbing ==
BEGIN
 \lineskip :=L 0pt
 \> == \@rtab
 \< == \@ltab
 \= == \@settab
 \+ == \@tabplus
 \- == \@tabminus
 \‘ == \@tabrj
 \’ == \@tablab
 \\ == BEGIN \@stopline \@startline END
 \\[DIST] == BEGIN
 \@stopline \vskip DIST \@startline\ignorespaces
 END
 * == BEGIN \@stopline \penalty 10000 \@startline END
 *[DIST] == BEGIN \@stopline \penalty 10000 \vskip DIST
 \@startline\ignorespaces END
 \@hightab := \@nxttabmar :=G \@firsttab
 \@tabpush :=G 0
 \dimen\@firsttab := \@totalleftmargin
 @rjfield :=G F
 \trivlist \item\relax
 if @minipage = F then \vskip \parskip fi
 \box\@tabfbox = \rlap{\indent\the\everypar}
 % note: \the\everypar sets @inlabel :=G F
 \@itemfudge == BEGIN \box\@tabfbox END
 \@startline
 \ignorespaces
 END

\@endtabbing ==
BEGIN
 \@stopline
 if \@tabpush > 0 then error message: "unmatched \poptabs" fi
 \endtrivlist
 END

\@rtab ==
BEGIN
 \@stopfield
 \@addfield
 if \@curtab < \@hightab
 then \@curtab :=G \@curtab + 1
 else error message "Undefined Tab" fi

```

```

\@tempdima := \dimen\@curtab - \dimen\@curtabmar
 - width of box \@curline
\box\@curline :=G \hbox{\unhbox\@curline + \hskip\@tempdima}
\@startfield
END

\@settab ==
BEGIN
 \@stopfield
 \@addfield
 if \@curtab < \@maxtab
 then \@curtab :=G \@curtab+1
 else error message: "Too many tabs" fi
 if \@curtab > \@hightab
 then \@hightab :=L \@curtab fi
 \dimen\@curtab :=L \dimen\@curtabmar + width of \box\@curline
 \@startfield
END

\@ltab ==
BEGIN
 \@ifatmargin
 then if \@curtabmar > \@firsttab
 then \@curtab :=G \@curtab - 1
 \@curtabmar :=G \@curtabmar - 1
 else error message "Too many untab" fi
 else error message "Left tab in middle of line"
 fi
END

\@tabplus ==
BEGIN
 if \@nxttabmar < \@hightab
 then \@nxttabmar :=G \@nxttabmar+1
 else error message "Undefined tab"
 fi
END

\@tabminus ==
BEGIN
 if \@nxttabmar > \@firsttab
 then \@nxttabmar :=G \@nxttabmar-1
 else error message "Too many untab"
 fi
END

\@tabrj ==
BEGIN \@stopfield
 \@addfield
 @rjfield :=G T

```

```

 \@startfield
 END

\@tablab ==
 BEGIN \@stopfield
 \box\@curline G:= \hbox{\box\@curline %% 'G' added 17 Jun 86
 \hskip - width of \box\@curfield
 \hskip -\tabbingsep
 \box\@curfield
 \hskip \tabbingsep }

 \@startfield
 END

\pushtabs ==
 BEGIN
 \@stopfield
 \@tabpush :=G \@tabpush + 1
 \begingroup
 \@contfield
 END

\poptabs ==
 BEGIN
 \@stopfield
 if \@tabpush > 0
 then \endgroup
 \@tabpush :=G \@tabpush - 1
 else error message: "Too many \poptabs"
 fi
 \@contfield
 END

```

- \a The accents \‘, \’ , and \= that have been redefined inside a tabbing environment can be called by typing \a‘ , \a’ , and \a=. The macro \a is defined in ltoutenc.dtx.

The ‘2ekernel’ code ensures that a \usepackage{autotabg} is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

```
1 <2ekernel>\expandafter\let\csname ver@autotabg.sty\endcsname\fmtversion
```

```

\@firsttab
\@maxtab
2 (*2ekernel)
3 \newdimen\@gtempa
4 \chardef\@firsttab=\the\allocationnumber
5 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
6 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
7 \newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
8 \newdimen\@gtempa
9 \chardef\@maxtab=\the\allocationnumber
10 \dimen\@firsttab=0pt

```

```

\@nxttabmar
\@curtabmar 11 \newcount\@nxttabmar
 \@curtab 12 \newcount\@curtabmar
 \@hightab 13 \newcount\@curtab
 \@tabpush 14 \newcount\@hightab
 15 \newcount\@tabpush

 \@curline
\@curfield 16 \newbox\@curline
\@tabfbox 17 \newbox\@curfield
 18 \newbox\@tabfbox

\if@rjfield
 19 \newif\if@rjfield

\@startline It is, in some sense, an error if the current margin tab setting is higher than
the value of \@hightab (which is a local variable). That this is allowed is a
fundamental design flaw which is not going to be corrected now.
 20 \gdef\@startline{%
 21 \ifnum \@nxttabmar >\@hightab
 22 \@badtab
 23 \global\@nxttabmar \@hightab
 24 \fi
 25 \global\@curtabmar \@nxttabmar
 26 \global\@curtab \@curtabmar
 27 \global\setbox\@curline \hbox {}%
 28 \@startfield
 29 \strut}

\@stopline
 30 \gdef\@stopline{%
 31 \unskip
 32 \@stopfield
 33 \if@rjfield
 34 \global\@rjfieldfalse
 35 \@tempdima\@totalleftmargin
 36 \advance\@tempdima\linewidth
 37 \hb@xt@\@tempdima{%
 38 \@itemfudge\hskip\dimen\@curtabmar
 39 \box\@curline
 40 \hfil
 41 \box\@curfield}%
 42 \else
 43 \@addfield
 44 \hbox{\@itemfudge\hskip\dimen\@curtabmar\box\@curline}%
 45 \fi}

\@startfield
 46 \gdef\@startfield{%
 47 \global\setbox\@curfield\hbox\bgroup\color@begingroup}

\@stopfield
 48 \gdef\@stopfield{%
 49 \color@endgroup\egroup}

```

```

\@contfield
50 \gdef\@contfield{%
51 \global\setbox\@curfield\hbox\bgroup\color@begingroup
52 \unhbox\@curfield}

\@addfield
53 \gdef\@addfield{\global\setbox\@curline\hbox{\unhbox
54 \@curline\unhbox\@curfield}}

\@ifatmargin
55 \gdef\@ifatmargin{\ifdim \wd\@curline =\z@}

\@tabcr
56 \gdef\@tabcr{\@stopline \@ifstar{\penalty \@M \@xtabcr}\@xtabcr}

\@xtabcr
57 \gdef\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}

\@itabcr
58 \gdef\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
59 \gdef\kill{\@stopfield\@startline\ignorespaces}

\tabbing We use \relax to prevent \item from scanning too far.
60 \gdef\tabbing{\lineskip \z@skip\let\>\@rtab\let\<\@ltab\let\=\@settab
61 \let\+\@tabplus\let\-\@tabminus\let\'\@tabrj\let\''\@tablab
62 \let\=\@tabcr
63 \@hightab\@firsttab
64 \global\@nxttabmar\@firsttab
65 \dimen\@firsttab\@totalleftmargin
66 \global\@tabpush\z@ \global\@rjfieldfalse
67 \trivlist \item\relax
68 \if@minipage\else\vskip\parskip\fi

69 \setbox\@tabfbox\hbox{%
70 \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
71 \def\@itemfudge{\box\@tabfbox}%
72 \@startline\ignorespaces}

\endtabbing
73 \gdef\endtabbing{%
74 \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}

\@rtab Omitted \global added to \@rtab 17 Jun 86
75 \gdef\@rtab{\@stopfield\@addfield\ifnum \@curtab<\@hightab
76 \global\advance\@curtab \@one \else\@badtab\fi
77 \@tempdima\dimen\@curtab
78 \advance\@tempdima -\dimen\@curtabmar
79 \advance\@tempdima -\wd\@curline
80 \global\setbox\@curline\hbox{\unhbox\@curline\hskip\@tempdima}%
81 \@startfield\ignorespaces}

```



```

\@settab
82 \gdef\@settab{\@stopfield\@addfield
83 \ifnum \@curtab <\@maxtab
84 \ifnum\@curtab =\@hightab
85 \advance\@hightab \@ne
86 \fi
87 \global\advance\@curtab \@ne
88 \else
89 \latexerror{Tab overflow}\@ehd
90 \fi
91 \dimen\@curtab \dimen\@curtabmar
92 \advance\dimen\@curtab \wd\@curline
93 \@startfield
94 \ignorespaces}

\@ltab
95 \gdef\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firsttab
96 \global\advance\@curtab \m@ne \global\advance\@curtabmar\m@ne\else
97 \@badtab\fi\else
98 \latexerror{string<\space in mid line}\@ehd\fi\ignorespaces}

\@tabplus
99 \gdef\@tabplus{%
100 \ifnum\@nxttabmar<\@hightab
101 \global\advance\@nxttabmar\@ne
102 \else
103 \@badtab
104 \fi
105 \ignorespaces}

\@tabminus
106 \gdef\@tabminus{%
107 \ifnum\@nxttabmar>\@firsttab
108 \global\advance\@nxttabmar\m@ne
109 \else
110 \@badtab
111 \fi
112 \ignorespaces}

\@tabrj
113 \gdef\@tabrj{%
114 \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}

\@tablab \setbox\@curline made \global in \@tablab. 17 Jun 86
115 \gdef\@tablab{%
116 \@stopfield
117 \global\setbox\@curline\hbox{%
118 \box\@curline
119 \hskip-\wd\@curfield \hskip-\tabbingsep
120 \box\@curfield
121 \hskip\tabbingsep}%
122 \@startfield
123 \ignorespaces}

```

`\pushtabs`

```
124 \gdef\pushtabs{%
125 \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
126 \@contfield}
```

`\poptabs` It is, in some sense, an error if, after the endgroup, the current tab setting is higher than the new value of `\@hightab` (which is a local variable). That this is allowed is a fundamental design flaw which is not going to be corrected now.

```
127 \gdef\poptabs{\@stopfield\@addfield
128 \ifnum \@tabpush >\z@
129 \endgroup
130 \global\advance\@tabpush \m@ne
131 \ifnum \@curtab >\@hightab
132 \global \@curtab \@hightab
133 \@badtab
134 \fi
135 \else
136 \@badpoptabs
137 \fi
138 \@contfield}
```

`\tabbingsep`

```
139 \newdimen\tabbingsep
```

## 56.2 array and tabular environments

ARRAY PARAMETERS:

`\arraycolsep`  
: half the width separating columns in an array environment

`\tabcolsep`  
: half the width separating columns in a tabular environment

`\arrayrulewidth`  
: width of rules

`\doublerulesep`  
: space between adjacent rules in array or tabular

`\arraystretch`  
: line spacing in array and tabular environments is done by placing a strut in every row of height and depth `\arraystretch` times the height and depth of the strut produced by an ordinary `\strut` command.

PREAMBLE:

The PREAMBLE argument of an array or tabular environment can contain the following:

`l,r,c` : indicate where entry is to be placed.  
`|` : for vertical rule  
`@{EXP}` : inserts the text EXP in every column.  
`\arraycolsep` or `\tabcolsep` spacing is suppressed.  
`*{N}{PRE}` : equivalent to writing N copies of PRE in the preamble.  
PRE may contain `*{N'}{EXP'}` expressions.  
`p{LEN}` : makes entry in parbox of width LEN.

SPECIAL ARRAY COMMANDS:

`\multicolumn{N}{FORMAT}{ITEM}` : replaces the next N column items by ITEM, formatted according to FORMAT. FORMAT should contain at most one l,r or c. If it contains none, then ITEM is ignored.

`\vline` : draws a vertical line the height of the current row. May appear in an array element entry.

`\hline` : draws a horizontal line between rows. Must appear either before the first entry (to appear above the first row) or right after a `\` command. If followed by another `\hline`, then adds a `\vskip` of `\doublerulesep`.

`\cline[i-j]` : draws horizontal lines between rows covering columns i through j, inclusive. Multiple commands may follow one another to provide lines covering several disjoint columns

`\extracolsep{WIDTH}` : for use inside an @ in the preamble. Causes a WIDTH space to be added between columns for the rest of the columns. This is in addition to the ordinary intercolumn space.

```
\array ==
 BEGIN
 \@acol == \@arrayacol
 \@classz == \@arrayclassz
 \@classiv == \@arrayclassiv
 \ == \@arraycr
 \@halignto == NULL
 \@tabarray
 END
```

```
\endarray{NAME} == BEGIN \crrc }} END
```

```
\tabular ==
 BEGIN
 \@halignto == NULL
 \@tabular
 END
```

```
\tabular*{WIDTH} ==
 BEGIN
 \@halignto == to WIDTH
 \@tabular
 END
```

```
\@tabular ==
 BEGIN
 \leavevmode
```

```

\hbox { $
 \@acol == \@tabacol
 \@classz == \@tabclassz
 \@classiv == \@tabclassiv
 \\ == \@tabularcr
 \@tabarray
END

\endtabular == BEGIN \crrc}} $} END

\@tabarray == if next char = [then \@array else \@array[c] fi

\@array[POS]{PREAMBLE} ==
BEGIN
 define \@arstrutbox to make \@arstrut produce strut of height
 and depth \arraystretch times the height and
 depth of a normal strut.
 \@mkpream{PREAMBLE}
 \@preamble == \halign \@halignto {\@tabskip=0pt\@arstrut
 eval{\@preamble}\@tabskip = 0pt\cr %%}
 \@startpbox == \@@startpbox
 \@endpbox == \@@endpbox
 if POS = t then \vtop
 else if POS = b then \vbox
 else \vcenter
 fi
 fi
 {
 \par ==L {} % changed 92/09/18
 \@sharp == #
 \protect == \relax
 \lineskip :=L 0pt
 \baselineskip :=L 0pt
 \@preamble
END

\@arraycr ==
BEGIN
 $ %% Prevents extra space at end of row's last entry.
 if next char = [
 then \@argarraycr
 else $ \cr %% Needed to balance $
 fi
END

\@argarraycr[LENGTH] ==
BEGIN
 $ %% Needed to balance $ of \@arraycr
 if LENGTH > 0
 then \@tempdima := depth of \@arstrutbox + LENGTH
 \vrule height 0pt width 0pt depth \@tempdima
 fi
 fi

```

```

 \cr
 else \cr \noalign{\vskip LENGTH}
 END

 \@tabularcr and \@argtabularcr same as \@arraycr and
 \@argarraycr
 except without the extra $'s.

\extracolsep
140 \def\extracolsep#1{\tabskip #1\relax}

 \array
141 \def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
142 \let\@classiv\@arrayclassiv
143 \let\\\@arraycr\let\@halignto\@empty\@tabarray}

 \endarray
\endtabular
144 \def\endarray{\crcr\egroup\egroup}
\endtabular*
145 \def\endtabular{\crcr\egroup\egroup $\egroup}
146 \expandafter \let \csname endtabular*\endcsname = \endtabular

 \tabular
147 \def\tabular{\let\@halignto\@empty\@tabular}

\tabular* Note that the change to use \setlength slightly alters the timing of the expansion
and use of the length in #1 but this is very unlikely to have any practical effect.
148 \namedef{tabular*}#1{%
149 \setlength\dimen@{#1}%
150 \edef\@halignto{to\the\dimen@}\@tabular}

\@tabular
151 \def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
152 \let\@classz\@tabclassz
153 \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}

\@tabarray RmS 91/11/04 added \m@th.
154 \def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}

 RmS 1993/11/03 changed \halign to \ialign and removed superfluous
 \tabskip assignment

\@array
155 \def\@array[#1]#2{%
156 \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
157 \bgroup

```

This next bit of code sets up the strut and then builds the halign and its preamble according to the specification in the second argument.

This code has been moved inside the box. A side effect of this has been to expose what was a buglet in the previous version: since the \@arstrut below is expanded and contains an \ifmmode then it could produce an unnecessary extra box in every row, thus wasting 'lots of' main memory.

```

158 \setbox\@arstrutbox\hbox{%
159 \vrule \@height\arraystretch\ht\strutbox
160 \@depth\arraystretch \dp\strutbox
161 \@width\z0}%
162 \@mkpream{#2}%
163 \edef\@preamble{%
164 \ialign \noexpand\@halignto
165 \bgroup \@arstrut \@preamble \tabskip\z@skip \cr}%

```

That is the end of setting up the preamble; now we reset things before executing the `\halign` built-up in `\@preamble`. The restorations could be done by introducing an extra group, thus saving tokens.

```

166 \let\@startpbox\@startpbox \let\@endpbox\@endpbox
167 \let\@tabularnewline\@%
168 \let\@par\@empty
169 \let\@sharp##%
170 \set@typeset@protect
171 \lineskip\z@skip\baselineskip\z@skip

```

If the parsing of the preamble goes wrong there may be some characters left which  $\TeX$  then tries to typeset, i.e., we would be in horizontal mode. That would produce an endless loop because the `\halign` expects vertical mode thus issues a `\par` but that is a no-op at this point. So we better test this case issue some error message and make a crude recovery by ending that horizontal mode with force. A better fix would be to ensure that we never pick up more than a single character token (not done).

```

172 \ifhmode \@preamerr\z@ \@@par\fi
173 \@preamble}

```

`\@arraycr` Array version of `\@`.

```

174 \def\@arraycr{%
175 ${\ifnum0=} \fi \ifstar \@xarraycr \@arraycr}

```

`\@arraycr`

```

176 \def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0={\fi}$}\cr}}

```

`\@argarraycr`

```

177 \def\@argarraycr[#1]{%
178 \ifnum0={\fi}$}\ifdim #1>\z@ \@xargarraycr{#1}\else
179 \@yargarraycr{#1}\fi}

```

`\@tabularnewline` Tabular version of `\@`.

```

180 \let\@tabularnewline\relax

```

`\@tabularcr`

```

181 \def\@tabularcr{%
182 {\ifnum0=} \fi \ifstar \@xtabularcr \@tabularcr}

```

`\@xtabularcr`

```

183 \def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0={\fi}\cr}}

```

`\@argtabularcr`

```

184 \def\@argtabularcr[#1]{%

```

```

185 \ifnum0='{\fi}%
186 \ifdim #1>\z@
187 \unskip\@xargarraycr{#1}%
188 \else
189 \@yargarraycr{#1}%
190 \fi}

```

\@xargarraycr

```

191 \def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
192 \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}

```

\@yargarraycr

```

193 \def\@yargarraycr#1{\cr\noalign{\vskip #1}}

```

```

\multicolumn \multicolumn{NUMBER}{FORMAT}{ITEM} ==
BEGIN
\multispan{NUMBER}
\beginngroup
\@addamp == null
\@mkpream{FORMAT}
\@sharp == ITEM
\protect == \relax
\@startpbox == \@startpbox
\@endpbox == \@endpbox
\@arstrut
\@preamble
\endngroup
END

```

The command `\def\@addamp{}` was removed from `\multicolumn` on 6 Dec 86 because it caused embedded array environments not to work. I think that it was included originally to prevent an error message if the 2nd argument to the `\multicolumn` command had two column specifiers.

8 Feb 89 — `\hbox{}` added after `\@preamble` to correct bug that occurred if `\multicolumn` preceded `\\[D]` with  $D > 0$ , caused by `\\[]` command doing an `\unskip`, which removed `\tabcolsep` glue inserted by `\multicolumn`.

This has been made long so that, for example, a p-column can contain multiple paragraphs; maybe the arguments of @-expressions should also be able to contain multiple paragraphs.

```

194 \long\def\multicolumn#1#2#3{\multispan{#1}\beginngroup
195 \@mkpream{#2}%
196 \def\@sharp{#3}\set@typeset@protect
197 \let\@startpbox\@startpbox\let\@endpbox\@endpbox
198 \@arstrut \@preamble\hbox{\endgroup\ignorespaces}

```

Codes for classes and character numbers of array, tabular and multicolumn arguments.

| Character | Class | Number |
|-----------|-------|--------|
| c         | 0     | 0      |
| l         | 0     | 1      |

|         |   |   |
|---------|---|---|
| r       | 0 | 2 |
|         | 1 | - |
| @       | 2 | - |
| p       | 3 | - |
| {@-exp} | 4 | - |
| {p-arg} | 5 | - |

`\@testpach \foo` : expands `\foo`, which should be an array parameter token, and sets `\@chclass` and `\@chnum` to its class and number. Uses `\@lastchclass` to distinguish 4 and 5

Preamble error codes

0: 'illegal character'  
 1: 'Missing @-exp'  
 2: 'Missing p-arg'

```
\@addamp ==
 BEGIN if \@firstamp = true then \@firstamp := false
 else &
 fi
 END
```

```
\@mkpream TOKENLIST ==
 BEGIN
 \@firstamp := T
 \@lastchclass := 6
 \@preamble == null
 \@sharp == \relax
 \@protect == BEGIN \noexpand\protect\noexpand END
 \@startpbox == \relax
 \@endpbox == \relax
 \@expast{TOKENLIST}
 for \@nextchar := expand(\reserved@a)
 do \@testpach{\@nextchar}
 case of \@chclass
 0 -> \@classz
 1 -> \@classi
 ...
 5 -> \@classv
 end case
 \@lastchclass := \@chclass
 od
 case of \@lastchclass
 0 -> \hskip \arraycolsep % lrc
 1 -> % |
 2 -> \@preamerr1 % 'Missing @-exp' % @
 3 -> \@preamerr2 % 'Missing p-arg' % p
 4 -> % @-exp
 5 -> \hskip \arraycolsep % p-exp
 end case
```



```

END

\@arrayclassz ==
BEGIN
 \@preamble := \@preamble *
 case of \@lastchclass
\arraycolsep
 0 -> \hskip \arraycolsep \@addamp \hskip
 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 \@arrayclassv except without the $... $

\@classv ==
BEGIN
 \@preamble :=
 \@preamble * \@startpbox{\@nextchar}\ignorespaces\@sharp
 \@endpbox
END

\@expast{S}:
Sets \reserved@a := S with all instances of *{N}{STRING}
replaced by N copies of STRING, where N > 0. An *
appearing inside braces is ignored, but *-expressions
inside STRING are expanded, so nested *-expressions are
handled properly.

\@expast{S} == BEGIN \@expast S *0x\@@ END

\@expast S1 *{N}{S2} S3 \@@ ==
BEGIN
 \reserved@a := S1
 \@tempcnta := N
 if \@tempcnta > 0
 then while \@tempcnta > 0 do \reserved@a := \reserved@a S2
 \@tempcnta := \@tempcnta - 1 od
 \reserved@b == \@expast
 else \reserved@b == \@xexnoop
 fi
 \expandafter \reserved@b \reserved@a S3 \@@
END

```

```

\@exnoop
199 \def\@exnoop #1\@{ }

\@expast
200 \def\@expast#1{\@expast #1*0x\@{ }

\@expast
201 \def\@expast#1*#2#3#4\@{ %
202 \edef\reserved@a{#1}%
203 \@tempcnta#2\relax
204 \ifnum\@tempcnta>\z@
205 \whilenum\@tempcnta>\z@\do
206 {\edef\reserved@a{\reserved@a#3}\advance\@tempcnta \m@ne}%
207 \let\reserved@b\@expast
208 \else
209 \let\reserved@b\@exnoop
210 \fi
211 \expandafter\reserved@b\reserved@a #4\@{ }

\if@firstamp
\@addamp 212 \newif\if@firstamp
213 \def\@addamp{%
214 \if@firstamp
215 \@firstampfalse
216 \else
217 \edef\@preamble{\@preamble &}%
218 \fi}

\@arrayacol
\@tabacol 219 \def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\@ampacol 220 \def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\@acolampacol 221 \def\@ampacol{\@addamp \@acol}
222 \def\@acolampacol{\@acol\@addamp\@acol}

\@mkpream
223 \def\@mkpream#1{\@firstampttrue\@lastchclass6
224 \let\@preamble\@empty
225 \let\protect\unexpandable@protect
226 \let\@sharp\relax
227 \let\@startpbox\relax\let\@endpbox\relax
228 \@expast{#1}%
229 \expandafter\@tfor \expandafter
230 \@nextchar \expandafter:\expandafter=\reserved@a\do
231 {\@testpach\@nextchar
232 \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
233 \or \@classiv \or \@classv \fi\@lastchclass\@chclass}%
234 \ifcase \@lastchclass \@acol
235 \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi}

\@arrayclassz
236 \def\@arrayclassz{\ifcase \@lastchclass \@acolampacol \or \@ampacol \or
237 \or \or \@addamp \or
238 \@acolampacol \or \@firstampfalse \@acol \fi}

```

```

239 \edef\@preamble{\@preamble
240 \ifcase \@chnum
241 \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
242 \or \hfil$\relax\@sharp$\fi}}

\@tabclassz RmS 91/08/14 inserted extra braces around entry for NFSS
243 \def\@tabclassz{%
244 \ifcase\@lastchclass
245 \@acolampacol
246 \or
247 \@ampacol
248 \or
249 \or
250 \or
251 \@addamp
252 \or
253 \@acolampacol
254 \or
255 \@firstampfalse\@acol
256 \fi
257 \edef\@preamble{%
258 \@preamble{%
259 \ifcase\@chnum
260 \hfil\ignorespaces\@sharp\unskip\hfil
261 \or
262 \hskip1sp\ignorespaces\@sharp\unskip\hfil
263 \or
264 \hfil\hskip1sp\ignorespaces\@sharp\unskip
265 \fi}}}

\@classi
266 \def\@classi{%
267 \ifcase\@lastchclass
268 \@acol\@arrayrule
269 \or
270 \@addtopreamble{\hskip \doublerulesep}\@arrayrule
271 \or
272 \or
273 \or
274 \@arrayrule
275 \or
276 \@acol\@arrayrule
277 \or
278 \@arrayrule
279 \fi}

\@classii
280 \def\@classii{%
281 \ifcase\@lastchclass
282 \or
283 \@addtopreamble{\hskip .5\arrayrulewidth}%
284 \fi}

```

```

\@classiii
285 \def\@classiii{\ifcase \@lastchclass \@acolampacol \or
286 \@addamp\@acol \or
287 \or \or \@addamp \or
288 \@acolampacol \or \@ampacol \fi}

\@tabclassiv
289 \def\@tabclassiv{\@addtopreamble\@nextchar}

\@arrayclassiv
290 \def\@arrayclassiv{\@addtopreamble{\$\@nextchar$}}

\@classv
291 \def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
292 \@sharp\@endpbox}}

\@addtopreamble
293 \def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}

\@chclass
\@lastchclass 294 \newcount\@chclass
\@chnum 295 \newcount\@lastchclass
296 \newcount\@chnum

\arraycolsep
\@tabcolsep 297 \newdimen\arraycolsep
\arrayrulewidth 298 \newdimen\@tabcolsep
\@doublerulesep 299 \newdimen\arrayrulewidth
300 \newdimen\@doublerulesep

\arraystretch
301 \def\arraystretch{1} % Default value.

\@arstrutbox
\@arstrut 302 \newbox\@arstrutbox
303 \def\@arstrut{%
304 \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}

\@arrayrule
305 \def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
306 \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}

\@testpatch
307 \def\@testpach#1{\@chclass \ifnum \@lastchclass=\tw@ 4 \else
308 \ifnum \@lastchclass=3 5 \else
309 \z@ \if #1c\@chnum \z@ \else
310 \if #1l\@chnum \@ne \else
311 \if #1r\@chnum \tw@ \else
312 \@chclass \if #1|\@ne \else
313 \if #1@\tw@ \else
314 \if #1p3 \else \z@ \@preamerr 0\fi
315 \fi \fi \fi \fi \fi \fi
316 \fi}

```

```

\hline
317 \def\hline{%
318 \noalign{\ifnum0='}\fi\hrule \@height \arrayrulewidth \futurelet
319 \reserved@a\@xhline}

\@xhline
320 \def\@xhline{\ifx\reserved@a\hline
321 \vskip\doublerulesep
Measure from the middle of the rules.
322 \vskip-\arrayrulewidth
323 \fi
324 \ifnum0='{ \fi}}

\vline
325 \def\vline{\vrule \@width \arrayrulewidth}

\cline The old LATEX2.09 implementation of \cline used up quite a lot of memory and
\@cline two precious count registers. This new (1995/09/14) implementation does not use
any count registers. It is coded in a way that depends heavily on the definition of
\multispan so that command has been moved here from the file ltplain.dtx.
These counters are no longer declared.

\newcount\@cla
\newcount\@clb

326 \def\cline#1{\@cline#1\@nil}

327 \def\@cline#1-#2\@nil{%
328 \omit

Use the counter from \multispan.
329 \@multicnt#1%
330 \advance\@multispan\m@ne
331 \ifnum\@multicnt=\@ne\@firstofone{&\omit}\fi
332 \@multicnt#2%
333 \advance\@multicnt-#1%
334 \advance\@multispan\@ne

The original had \unskip at this point, but how could a skip get here ???
335 \leaders\hrule\@height\arrayrulewidth\hfill
336 \cr

This is back spacing is fairly horrible, but it is what happened in the old version. . .
An alternative would be to make \cline look ahead for a following \cline as does
\hline. This would alter the spacing in existing documents so keep the old version
in the kernel. Perhaps a package should do this differently.
337 \noalign{\vskip-\arrayrulewidth}}

\mscount The \mscount counter is no longer declared, saving a csname and a register. It is
declared in compatibility mode.

\multispan Modify \multispan slightly from its plain TEX definition to allow more efficient
\@multispan code sharing with \multicolumn. Also share a count register with \multiput.
\sp@n 338 \def\multispan{\omit\@multispan}

```

```

339 \def\@multispan#1{%
340 \@multicnt#1\relax
341 \loop\ifnum\@multicnt>\@ne \sp@n\repeat}
342 \def\sp@n{\span\omit\advance\@multicnt\m@ne}

\@startpbox Helper macros for 'p' columns.
\@endpbox \@startpbox{<width>} text \egroup is essentially \parbox{<width>}{<text>}
 \@endpbox is essentially \unskip \strut \par \egroup\hfil (Changed 14
 Jan 89) (changed again 1994/05/13)
343 \def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
344 \def\@endpbox{\@finalstrut\@arstrutbox\par\egroup\hfil}

 14 Jan 89: Def of \@endpbox changed from
 \def\@endpbox{\par\vskip\dp\@arstrutbox\egroup\hfil}
 so vertical spacing works out right if the last line of a 'p' entry has a descender.

\@@startpbox
\@@endpbox 345 \let\@@startpbox=\@startpbox
 346 \let\@@endpbox=\@endpbox

347 </2ekernel>

```

# File D

## ltpictur.dtx

### 57 Picture Mode

Picture mode commands. In addition to the commands available in L<sup>A</sup>T<sub>E</sub>X2.09, This section adds the new `\qbezier` command for drawing curves.

`\qbezier` `\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.

```

\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.

The ‘2ekernel’ code ensures that a `\usepackage{autopict}` is essentially ignored if a ‘full’ format is being used that has picture mode already in the format.

```
1 (2ekernel)\expandafter\let\csname ver@autopict.sty\endcsname\fmtversion
```

```

\@wholewidth
\@halfwidth 2 (*2ekernel)
 3 \newdimen\@wholewidth
 4 \newdimen\@halfwidth

\unitlength
 5 \newdimen\unitlength \unitlength =1pt

\@picbox
\@picht 6 \newbox\@picbox
 7 \newdimen\@picht

\picture #1 should be white space.

\pictur@ #1 should be a ((eating any white space before the bracket),
 8 \long\gdef\picture#1{\pictur@#1}
 9 \gdef\pictur@(#1){%
 10 \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}

```

```

\@picture
11 \gdef\@picture(#1,#2)(#3,#4){%
12 \@picht#2\unitlength
13 \setbox\@picbox\hb@xt@#1\unitlength\bgroup
14 \hskip -#3\unitlength
15 \lower #4\unitlength\hbox\bgroup
16 \ignorespaces}

\endpicture
17 \gdef\endpicture{%
18 \egroup\hss\egroup
19 \ht\@picbox\@picht\dp\@picbox\z@
20 \mbox{\box\@picbox}}

In the definitions of \put and \multiput, \hskip was replaced by \kern just
in case arg #3 = "plus". (Bug detected by Don Knuth. changed 20 Jul 87).

21 \long\gdef\put(#1,#2)#3{%
22 \@killglue\raise#2\unitlength
23 \hb@xt@\z@\{\kern#1\unitlength #3\hss}%
24 \ignorespaces}

\multiput #3 had better be a (.
25 \gdef\multiput(#1,#2)#3{%
26 \@xdim #1\unitlength
27 \@ydim #2\unitlength
28 \@multiput{)}

\multiput
29 \long\gdef\@multiput(#1,#2)#3#4{%
30 \@killglue\@multicnt #3\relax
31 \@whilenum \@multicnt >\z@\do
32 {\raise\@ydim\hb@xt@\z@\{\kern\@xdim #4\hss}%
33 \advance\@multicnt\m@ne
34 \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
35 \ignorespaces}

\@killglue
36 \gdef\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}

\thinlines
\thicklines
37 \gdef\thinlines{\let\@linefnt\tenln \let\@circlefnt\tencirc
38 \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
39 \gdef\thicklines{\let\@linefnt\tenlnw \let\@circlefnt\tencircw
40 \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}

\linethickness
41 \gdef\linethickness#1{\@wholewidth #1\relax \@halfwidth .5\@wholewidth}

\ishortstack
42 \gdef\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}

```

```

\@ishortstack
43 \gdef\@shortstack[#1]{%
44 \leavevmode
45 \vbox\bgroup
46 \baselineskip-\p@\lineskip 3\p@
47 \let\mb@l\hss\let\mb@r\hss
48 \expandafter\let\csname mb@#1\endcsname\relax
49 \let\\\@stackcr
50 \@ishortstack}

\@ishortstack
51 \gdef\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\cr}\egroup}

\@stackcr
\@ixstackcr
52 \gdef\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
53 \gdef\@ixstackcr{\@ifnextchar[\@istackcr{\cr\ignorespaces}}

\@istackcr
54 \gdef\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}

\line(X,Y){LEN} ==
BEGIN
\@xarg := X
\@yarg := Y
\@linelen := LEN * \unitlength
if \@xarg = 0
then \@vline
else if \@yarg = 0
then \@hline
else \@sline
if
if
END

\@sline ==
BEGIN
if \@xarg < 0
then @negarg := T
\@xarg := -\@xarg
\@yyarg := -\@yarg
else @negarg := F
\@yyarg := \@yarg
fi
\@tempcnta := |\@yyarg|
if \@tempcnta > 6
then error: 'LATEX ERROR: Illegal \line or \vector argument.'
\@tempcnta := 0
fi
\box\@linechar := \hbox{\@linefnt \@getlinechar(\@xarg,\@yyarg)}
}

```

```

if \@yarg > 0 then \@upordown = \raise
 \@clnht := 0
 else \@upordown = \lower
 \@clnht := height of \box\@linechar
fi
\@clnwd := width of \box\@linechar
if @negarg
 then \hskip - width of \box\@linechar
 \reserved@a == \hskip - 2* width of box \@linechar
 else \reserved@a == \relax
fi
%% Put out integral number of line segments
while \@clnwd < \@linelen
 do \@upordown \@clnht \copy\@linechar
 \reserved@a
 \@clnht := \@clnht + ht of \box\@linechar
 \@clnwd := \@clnwd + width of \box\@linechar
 od

%% Put out last segment
\@clnht := \@clnht - height of \box\@linechar
\@clnwd := \@clnwd - width of \box\@linechar
\@tempdima := \@linelen - \@clnwd
\@tempdimb := \@tempdima - width of \box\@linechar
if @negarg then \hskip -\@tempdimb
 else \hskip \@tempdimb
fi
\@tempdima := 1000 * \@tempdima
\@tempcnta := \@tempdima / width of \box\@linechar
\@tempdima := (\@tempcnta * ht of \box\@linechar)/1000
\@clnht := \@clnht + \@tempdima
if \@linelen < width of box\@linechar
 then \hskip width of box\@linechar
 else \hbox{\@upordown \@clnht \copy\@linechar}
fi
END

\@hline ==
BEGIN
 if \@xarg < 0 then \hskip -\@linelen \fi
 \vrule height \@halfwidth depth \@halfwidth width \@linelen
 if \@xarg < 0 then \hskip -\@linelen \fi
END

\@vline == if \@yarg < 0 \@downline else \@upline fi

\@getlinechar(X,Y) ==
BEGIN
 \@tempcnta := 8*X - 9

```

```

 if Y > 0
 then \@tempcnta := \@tempcnta + Y
 else \@tempcnta := \@tempcnta - Y + 64
 fi
 \char\@tempcnta
 END

\vector(X,Y){LEN} ==
BEGIN
 \@xarg := X
 \@yarg := Y
 \@linelen := LEN * \unitlength
 if \@xarg = 0
 then \vvector
 else if \@yarg = 0
 then \hvector
 else \svector
 fi
 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

```

\if@negarg

```
55 \newif\if@negarg
```

\line

```

56 \gdef\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
57 \@linelen #3\unitlength
58 \ifdim\@linelen<\z@\@badlinearg\else
59 \ifnum\@xarg =\z@ \@vline
60 \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
61 \fi
62 \fi}

```

\@sline

```

63 \gdef\@sline{%
64 \ifnum\@xarg<\z@ \@negargtrue \@xarg -\@xarg \@yyarg -\@yarg
65 \else \@negargfalse \@yyarg \@yarg \fi
66 \ifnum \@yyarg >\z@ \@tempcnta\@yyarg \else \@tempcnta -\@yyarg \fi

```

```

67 \ifnum \@tempcnta > 6 \badlinearg \@tempcnta \z@ \fi
68 \ifnum \@xarg > 6 \badlinearg \@xarg \one \fi
69 \setbox \@linechar \hbox{\@linefont\@getlinechar(\@xarg,\@yyarg)}%

```

If we have something like `\line(5,5){30}` the `\@linechar` will not contain a char and later on we will end in an infinite loop. So we check the width of the box and put in something as an emergency fix if necessary.

```

70 \ifdim \wd \@linechar = \z@
71 \setbox \@linechar \hbox{.}%
72 \badlinearg
73 \fi
74 \ifnum \@yarg > \z@ \let \@upordown \raise \@clnht \z@
75 \else \let \@upordown \lower \@clnht \ht \@linechar \fi
76 \@clnwd \wd \@linechar
77 \if@negarg
78 \hskip -\wd \@linechar \def \reserved@a {\hskip -2\wd \@linechar}%
79 \else
80 \let \reserved@a \relax
81 \fi
82 \@whiledim \@clnwd < \@linelen \do
83 {\@upordown \@clnht \copy \@linechar
84 \reserved@a
85 \advance \@clnht \ht \@linechar
86 \advance \@clnwd \wd \@linechar}%
87 \advance \@clnht -\ht \@linechar
88 \advance \@clnwd -\wd \@linechar
89 \@tempdima \@linelen \advance \@tempdima -\@clnwd
90 \@tempdimb \@tempdima \advance \@tempdimb -\wd \@linechar
91 \if@negarg \hskip -\@tempdimb \else \hskip \@tempdimb \fi
92 \multiply \@tempdima \@m
93 \@tempcnta \@tempdima
94 \@tempdima \wd \@linechar \divide \@tempcnta \@tempdima
95 \@tempdima \ht \@linechar \multiply \@tempdima \@tempcnta
96 \divide \@tempdima \@m
97 \advance \@clnht \@tempdima
98 \ifdim \@linelen < \wd \@linechar
99 \hskip \wd \@linechar

```

Warn if line gets so short that it can't be printed. But don't warn if it is exactly zero since that was probably deliberate (e.g., to get a vector head only).

```

100 \ifdim \@linelen = \z@
101 \else
102 \@picture@warn
103 \fi
104 \else \@upordown \@clnht \copy \@linechar \fi}

```

`\@hline`

```

105 \gdef \@hline{\ifnum \@xarg < \z@ \hskip -\@linelen \fi
106 \vrule \@height \@halfwidth \@depth \@halfwidth \@width \@linelen
107 \ifnum \@xarg < \z@ \hskip -\@linelen \fi}

```

`\getlinechar`

```

108 \gdef \@getlinechar(#1,#2){\@tempcnta #1 \relax \multiply \@tempcnta 8%
109 \advance \@tempcnta -9 \ifnum #2 > \z@ \advance \@tempcnta #2 \relax \else

```

```

110 \advance\@tempcnta -#2\relax\advance\@tempcnta 64 \fi
111 \char\@tempcnta}

\vector
112 \gdef\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
113 \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
114 \ifnum\@tempcnta<5\relax
115 \@linelen #3\unitlength
116 \ifdim\@linelen<\z@\@badlinearg\else
117 \ifnum\@xarg =\z@ \@vvector
118 \else \ifnum\@yarg =\z@ \@hvector \else \@svector\fi
119 \fi
120 \fi
121 \else\@badlinearg\fi}

\@hvector
122 \gdef\@hvector{\@hline\hb@xt@\z@\{\@linefnt
123 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
124 \hss\@getrarrow(1,0)\fi}}

\@vvector
125 \gdef\@vvector{\ifnum \@yarg <\z@ \@downvector \else \@upvector \fi}

\@svector
126 \gdef\@svector{\@sline
127 \@tempcnta\@yarg \ifnum\@tempcnta <\z@ \@tempcnta -\@tempcnta\fi
128 \ifnum\@tempcnta <5%
129 \hskip -\wd\@linechar
130 \@upordown\@clnht \hbox{\@linefnt \if@negarg
131 \@getlarrow(\@xarg,\@yyarg)\else \@getrarrow(\@xarg,\@yyarg)\fi}%
132 \else\@badlinearg\fi}

\@getlarrow
133 \gdef\@getlarrow(#1,#2){\ifnum #2=\z@ \@tempcnta 27 % '33
134 \else
135 \@tempcnta #1\relax\multiply\@tempcnta \sixt@@n
136 \advance\@tempcnta -9 \@tempcntb #2\relax\multiply\@tempcntb \tw@
137 \ifnum \@tempcntb >\z@ \advance\@tempcnta \@tempcntb
138 \else\advance\@tempcnta -\@tempcntb\advance\@tempcnta 64
139 \fi\fi\char\@tempcnta}

\@getrarrow
140 \gdef\@getrarrow(#1,#2){\@tempcntb #2\relax
141 \ifnum\@tempcntb <\z@ \@tempcntb -\@tempcntb\relax\fi
142 \ifcase \@tempcntb\relax \@tempcnta 45 % '55
143 \or
144 \ifnum #1<\thr@@ \@tempcnta #1\relax\multiply\@tempcnta
145 24\advance\@tempcnta -6 \else \ifnum #1=\thr@@ \@tempcnta 49
146 \else\@tempcnta 58 \fi\fi\or
147 \ifnum #1<\thr@@ \@tempcnta=#1\relax\multiply\@tempcnta
148 24\advance\@tempcnta -\thr@@ \else \@tempcnta 51 \fi\or
149 \@tempcnta #1\relax\multiply\@tempcnta

```



```

150 \sixt@n \advance\@tempcnta -\tw@ \else
151 \@tempcnta #1\relax\multiply\@tempcnta
152 \sixt@n \advance\@tempcnta 7 \fi\ifnum #2<\z@ \advance\@tempcnta 64 \fi
153 \char\@tempcnta}

\@vline
154 \gdef\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}

\@upline
155 \gdef\@upline{%
156 \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
157 \@height \@linelen \@depth \z@\hss}}

\@downline
158 \gdef\@downline{%
159 \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
160 \@height \z@ \@depth \@linelen \hss}}

\@upvector
161 \gdef\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}% '66
162 \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}

\@downvector
163 \gdef\@downvector{\@downline\lower \@linelen
164 \hb@xt@\z@{\@linefnt\char 63 % '77
165 \hss}}

\dashbox{D}(X,Y) ==
BEGIN
leave vertical mode
\hb@xt@ 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

```

```

\@makepicbox(X,Y)
END

```

\dashbox

```

166 \gdef\dashbox#1(#2,#3){\leavevmode\hb@xt@z@{\baselineskip \z@skip
167 \lineskip \z@skip
168 \@dashdim #2\unitlength
169 \@dashcnt \@dashdim \advance\@dashcnt 200
170 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
171 \ifodd\@dashcnt\@dashdim \z@
172 \advance\@dashcnt \@one \divide\@dashcnt \tw@
173 \else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
174 \advance\@dashcnt \m@ne
175 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
176 \@width \@dashdim}\put(0,0){\copy\@dashbox}%
177 \put(0,#3){\copy\@dashbox}%
178 \put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
179 \put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
180 \multiply\@dashdim \thr@@
181 \fi
182 \setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
183 \@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
184 \put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
185 \do{\copy\@dashbox\advance\@tempcnta \@one }}\@tempcnta\z@
186 \put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
187 \do{\copy\@dashbox\advance\@tempcnta \@one }}%
188 \@dashdim #3\unitlength
189 \@dashcnt \@dashdim \advance\@dashcnt 200
190 \@dashdim #1\unitlength\divide\@dashcnt \@dashdim
191 \ifodd\@dashcnt \@dashdim \z@
192 \advance\@dashcnt \@one \divide\@dashcnt \tw@
193 \else
194 \divide\@dashdim \tw@ \divide\@dashcnt \tw@
195 \advance\@dashcnt \m@ne
196 \setbox\@dashbox\hbox{\hskip -\@halfwidth
197 \vrule \@width \@wholewidth
198 \@height \@dashdim}\put(0,0){\copy\@dashbox}%
199 \put(#2,0){\copy\@dashbox}%
200 \put(0,#3){\lower\@dashdim\copy\@dashbox}%
201 \put(#2,#3){\lower\@dashdim\copy\@dashbox}%
202 \multiply\@dashdim \thr@@
203 \fi
204 \setbox\@dashbox\hbox{\vrule \@width \@wholewidth

```

```

205 \@height #1\unitlength}\@tempcnta\z@
206 \put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\dashcnt
207 \do{\vskip #1\unitlength\copy\dashbox\advance\@tempcnta \@ne }%
208 \vskip\dashdim}}\@tempcnta\z@
209 \put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\dashcnt
210 \do{\vskip #1\unitlength\copy\dashbox\advance\@tempcnta \@ne }%
211 \vskip\dashdim}}\@makepicbox(#2,#3)}

```

## CIRCLES AND OVALS

### USER COMMANDS:

`\circle{D}` : Produces the circle with the diameter as close as possible to  $D * \text{\unitlength}$ . `\put(X,Y){\circle{D}}` puts the circle with its center at (X,Y).

`\oval(X,Y)` : Makes an oval as round as possible that fits in the rectangle of width  $X * \text{\unitlength}$  and height  $Y * \text{\unitlength}$ . The reference point is the center.

`\oval(X,Y)[POS]` : 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 `\@ovxx`.

`\@getcirc {DIAM}` : Sets `\@tempcnta` to the character number of the top-right quarter circle with the largest diameter less than or equal to DIAM. Sets `\@tempboxa` to an hbox containing that character. Sets `\@tempdima` to `\wd \@tempboxa`, which is the distance from the circle's left outside edge to its right inside edge. (These characters are like those described in the

TeXbook, pp. 389-90.)

```
\@getcirc {DIAM} ==
BEGIN
 \@tempcnta := integer coercion of (DIAM + 2pt)
 + 2pt added 1 Nov 88
 \@tempcnta := \@tempcnta / integer coercion of 4pt
 if \@tempcnta > 10
 then \@tempcnta := 10 fi
 if \@tempcnta > 0
 then \@tempcnta := \@tempcnta-1
 else LaTeX Warning: Oval too small.
 fi
 \@tempcnta := 4 * \@tempcnta
 \@tempboxa := \hbox{\@circlefnt \char \@tempcnta}
 \@tempdima := \wd \@tempboxa
END

\@put{X}{Y}{OBJ} ==
BEGIN
 \raise Y \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

```

\if@ovt If producing the Top Bottom Left or Right of an oval.

\if@ovb 212 \newif\if@ovt

\if@ovl 213 \newif\if@ovb

\if@ovr 214 \newif\if@ovl

215 \newif\if@ovr

\@ovxx

\@ovyy 216 \newdimen\@ovxx

\@ovdx

\@ovdy

\@ovro File D: ltpictur.dtx Date: 2016/03/29 Version v1.11

\@ovri

```

217 \newdimen\@ovvy
218 \newdimen\@ovdx
219 \newdimen\@ovdy
220 \newdimen\@ovro
221 \newdimen\@ovri

```

\advance\@tempdima 2pt\relax added 1 Nov 88 to fix bug in which size of drawn circle not monotonic function of argument of \circle, caused by different rounding for dimensions of large and small circles.

\@getcirc

```

222 \gdef\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
223 \@tempcnta\@tempdima
224 \@tempdima 4\p@ \divide\@tempcnta\@tempdima
225 \ifnum \@tempcnta >10\relax
226 \picture@warn
227 \@tempcnta 10\relax
228 \fi
229 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
Warn if requirements for oval or circle can't be met.
230 \else \picture@warn \fi
231 \multiply\@tempcnta 4\relax
232 \setbox \@tempboxa \hbox{\@circlefont
233 \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.

```

234 \def\@picture@warn{\@latex@warning{
235 \string\oval, \string\circle, or \string\line\space
236 size unavailable}}

```

\@put

```

237 \gdef\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}

```

\@oval

```

238 \gdef\@oval(#1,#2){\ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2) []}}
239 \kernel
240 \latexrelease\IncludeInRelease{2016/03/31}%
241 \latexrelease \@ovhlinetrue}%
242 \latexrelease {Avoid almost zero length leaders}%
243 \kernel | \latexrelease

```

\if@ovvline Tests whether horizontal or vertical lines are needed.

```

\if@ovhline 244 \newif\if@ovvline \@ovvlinetrue
245 \newif\if@ovhline \@ovhlinetrue

```

\@oval

```

246 \gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
247 \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
248 \@ovvlinefalse \@ovhlinefalse

```



```

249 \@tfor\reserved@a :=#3\do{\csname @ov\reserved@a false\endcsname}%
250 \@ovxx #1\unitlength
251 \@ovyy #2\unitlength

252 \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
253 \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue \fi\fi

254 \advance \@tempdimb -2\p@
255 \@getcirc \@tempdimb
256 \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
257 \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
258 \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@

259 \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
260 \ifdim \@ovdy >\z@ \@ovvlinetrue \fi

261 \@circlefnt \setbox\@tempboxa
262 \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
263 \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
264 \if@ovt \@ovhorz \kern -\@ovxx \fi
265 \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
266 \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
267 \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
268 \endgroup

```

\@ovvert

```

269 \gdef\@ovvert#1#2{\vbox to\@ovyy{%
270 \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
271 \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
272 \else \kern \@ovri \kern \@ovdy \fi

273 \if@ovvline \leaders\vrule \@width \@wholewidth \fi

274 \vfil \nointerlineskip
275 \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
276 \hbox{\char \@tempcntb}%
277 \else \kern \@ovdy \kern \@ovro \fi}}

```

\@ovhorz

```

278 \gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
279 \if@ovr \else \kern \@ovdx \fi

280 \if@ovhline \leaders \hrule \@height \@wholewidth \fi

281 \hfil
282 \if@ovl \else \kern \@ovdx \fi
283 \kern \@ovri}}

284 </2kernel | latexrelease>
285 <latexrelease>\EndIncludeInRelease
286 <latexrelease>\IncludeInRelease{0000/00/00}%
287 <latexrelease> {\@ovhlinetrue}%
288 <latexrelease> {Avoid almost zero length leaders}%
289 <latexrelease>\let\if@ovvline\@undefined
290 <latexrelease>\let\if@ovhline\@undefined
291 <latexrelease>\gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
292 <latexrelease> \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue

```

```

293 <latexrelease> \@tfor\reserved@a :=#3\do
294 <latexrelease> {\csname @ov\reserved@a false\endcsname}%
295 <latexrelease> \@ovxx #1\unitlength
296 <latexrelease> \@ovyy #2\unitlength
297 <latexrelease> \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
298 <latexrelease> \advance \@tempdimb -2\p@
299 <latexrelease> \@getcirc \@tempdimb
300 <latexrelease> \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
301 <latexrelease> \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
302 <latexrelease> \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
303 <latexrelease> \@circlefnt \setbox\@tempboxa
304 <latexrelease> \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
305 <latexrelease> \if@ovl
306 <latexrelease> \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
307 <latexrelease> \fi
308 <latexrelease> \if@ovt \@ovhorz \kern -\@ovxx \fi
309 <latexrelease> \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
310 <latexrelease> \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
311 <latexrelease> \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
312 <latexrelease> \endgroup}
313 <latexrelease>\gdef\@ovvert#1#2{\vbox to\@ovyy{%
314 <latexrelease> \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
315 <latexrelease> \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
316 <latexrelease> \else \kern \@ovri \kern \@ovdy \fi
317 <latexrelease> \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
318 <latexrelease> \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
319 <latexrelease> \hbox{\char \@tempcntb}%
320 <latexrelease> \else \kern \@ovdy \kern \@ovro \fi}}
321 <latexrelease>\gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
322 <latexrelease> \if@ovr \else \kern \@ovdx \fi
323 <latexrelease> \leaders \hrule \@height \@wholewidth \hfil
324 <latexrelease> \if@ovl \else \kern \@ovdx \fi
325 <latexrelease> \kern \@ovri}}
326 <latexrelease>\EndIncludeInRelease
327 (*2kernel)

```

\circle

```
328 \gdef\circle{\@inmatherr\circle\@ifstar\@dot\@circle}
```

\@circle

```

329 \gdef\@circle#1{%
330 \begingroup \boxmaxdepth \maxdimen \@tempdimb #1\unitlength
331 \ifdim \@tempdimb >15.5\p@ \@getcirc\@tempdimb
332 \@ovro\ht\@tempboxa
333 \setbox\@tempboxa\hbox{\@circlefnt
334 \advance\@tempcnta\tw@ \char \@tempcnta
335 \advance\@tempcnta@m@ne \char \@tempcnta \kern -2\@tempdima
336 \advance\@tempcnta\tw@
337 \raise \@tempdima \hbox{\char\@tempcnta}\raise \@tempdima
338 \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
339 \@put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
340 \else \@circ\@tempdimb{96}\fi\endgroup}

```

\@dot Internal form of \circle\*.

```

341 \gdef\@dot#1{\@tempdimb #1\unitlength \@circ\@tempdimb{112}}

\@circ
342 \gdef\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
343 \@tempcnta\@tempdima \@tempdima \p@
344 \divide\@tempcnta\@tempdima
345 \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
346 \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
347 \advance\@tempcnta #2\relax
348 \@circlefnt \char\@tempcnta}

\@xarg Counters used for manipulating the ‘slope’ arguments.
\@yarg 349 \newcount\@xarg
\@yyarg 350 \newcount\@yarg
351 \newcount\@yyarg

\@multicnt Counter used in \multiput, and also \multicolumn.
352 \newcount\@multicnt

\@xdim Length registers.
\@ydim 353 \newdimen\@xdim
354 \newdimen\@ydim

\@linechar Box for holding a line segment character, for sloping lines.
355 \newbox\@linechar

\@linelen Length of the line currently being built.
356 \newdimen\@linelen

\@clnwd Height and width of current line segment.
\@clnht 357 \newdimen\@clnwd
358 \newdimen\@clnht

\@dashdim \dashbox internal registers.
\@dashbox 359 \newdimen\@dashdim
\@dashcnt 360 \newbox\@dashbox
361 \newcount\@dashcnt

Initialization: “\thinlines”
362 \let\@linefnt\tenln
363 \let\@circlefnt\tencirc
364 \@wholewidth\fontdimen8\tenln
365 \@halfwidth .5\@wholewidth

```

## 57.1 Curves

The new `\qbezier` command, based on the old `\bezier` defined in `bezier.sty`.

```

\qbezier[N] == \bezier{N}

\bezier{N}(AX,AY)(BX,BY)(CX,CY) ==
BEGIN

```

```

IF N = 0
 THEN \@xdima := |BX - AX|
 \@xb := |CX - BX|
 \@xa := Max(\@xa, \@xb)
 \@ya := |BY - AY|
 \@yb := |CY - BY|
 \@ya := Max(\@ya, \@yb)
 @sc := Max(\@xa, \@ya)
 %% The coefficient .5 below is the degree of overlap of
 %% successive points, where 1 is no overlap and 0 is
 %% complete overlap. A coefficient of C multiplies
 %% the number of points plotted by 1/C.
 %%
 \@xa := .5 * \@halfwidth
 @sc := @sc / \@halfwidth
 @sc := Max(@sc, qbeziermax)
 ELSE @sc := N
 @scp := @sc+1
 \@xb := 2 * (BX - AX) * \unitlength
 \@xa := ((CX-AX)*\unitlength - \@xb)/@sc
 \@yb := 2 * (BY - AY) * \unitlength
 \@ya := ((CY-AY)*\unitlength - \@yb)/@sc
 \@pictdot := square rule of width \@wholewidth
 \count@ := 0
 WHILE \count@ < @scp
 DO \@xdim := ((\count@*\@xa + @xb) / @sc) * \count@
 \@ydim := ((\count@*\@ya + @yb) / @sc) * \count@
 plot pt with relative coords (\@xdim,\@ydim)
 \count@ := \count@+1
 OD

```

`\qbeziermax` The maximum number of points to plot.

```
366 \gdef\qbeziermax{500}
```

In the code below, to save registers `\@a...` are not used. Instead other registers are reused.

```

\newcounter{@sc} -> \c@multicnt
\newcounter{@scp} -> \c@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 (.

```
367 \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 (.

```
368 \gdef\bezier#1)#2(#3)#4({\@bezier#1)(#3){}
```

\@bezier

```
369 \gdef\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
370 \ifnum #1=\z@
371 \@ovxx #4\unitlength
372 \advance\@ovxx -#2\unitlength
373 \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
374 \@ovdx #6\unitlength
375 \advance\@ovdx -#4\unitlength
376 \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
377 \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
378 \@ovyy #5\unitlength
379 \advance\@ovyy -#3\unitlength
380 \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
381 \@ovdy #7\unitlength
382 \advance\@ovdy -#5\unitlength
383 \ifdim \@ovdy<\z@ \@ovdy -\@ovdy \fi
384 \ifdim \@ovyy<\@ovdy \@ovyy \@ovdy \fi
385 \@multicnt
386 \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
387 \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
388 \ifnum \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax \fi
389 \else \@multicnt#1\relax \fi
390 \@tempcnta\@multicnt \advance\@tempcnta\@ne
391 \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
392 \multiply\@ovdx \tw@
393 \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
394 \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
395 \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
396 \multiply\@ovdy \tw@
397 \@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
398 \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt

399 \setbox\@tempboxa\hbox{%
400 \hskip -\@halfwidth
401 \vrule \@height\@halfwidth
402 \@depth \@halfwidth
403 \@width \@wholewidth}%
404 \put(#2,#3){%
405 \count@\z@
406 \@whilenum{\count@<\@tempcnta}\do
407 {\@xdim\count@\@ovxx
408 \advance\@xdim\@ovdx
409 \divide\@xdim\@multicnt
410 \multiply\@xdim\count@
411 \@ydim\count@\@ovyy
412 \advance\@ydim\@ovdy
413 \divide\@ydim\@multicnt
414 \multiply\@ydim\count@
415 \raise \@ydim
416 \hb@xt@\z@{\kern\@xdim
417 \unhcopy\@tempboxa\hss}%
418 \advance\count@\@ne}}
419 /2ekernel)
```

# File E

## ltthm.dtx

### 58 Theorem Environments

The user creates his own theorem-like environments with the command

```
\newtheorem{<name>}{<text>}[<counter>] or
\newtheorem{<name>}[<oldname>]{<text>}
```

This defines the environment  $\langle name \rangle$  to be just as one would expect a theorem environment to be, except that it prints  $\langle text \rangle$  instead of “Theorem”.

If  $\langle oldname \rangle$  is given, then environments  $\langle name \rangle$  and  $\langle oldname \rangle$  use the same counter, so using a  $\langle name \rangle$  environment advances the number of the next  $\langle name \rangle$  environment, and vice-versa.

If  $\langle counter \rangle$  is given, then environment  $\langle name \rangle$  is numbered within  $\langle counter \rangle$ .

E.g., if  $\langle counter \rangle = \text{subsection}$ , then the first  $\langle name \rangle$  in subsection 7.2 is numbered  $\langle text \rangle$  7.2.1.

The way  $\langle name \rangle$  environments are numbered can be changed by redefining  $\backslash\text{the}\langle name \rangle$ .

#### DOCUMENT STYLE PARAMETERS

$\backslash\text{thmcounter}\{\text{COUNTER}\}$  : A command such that  
 $\backslash\text{edef}\backslash\text{the}\text{COUNTER}\{\backslash\text{thmcounter}\{\text{COUNTER}\}\}$   
defines  $\backslash\text{the}\text{COUNTER}$  to produce a number for a theorem environment.  
The default is:  
BEGIN  $\backslash\text{noexpand}\backslash\text{arabic}\{\text{COUNTER}\}$  END

$\backslash\text{thmcountersep}$  : A separator placed between a theorem number and the number of the counter within which it is numbered.  
E.g., to make the third theorem of section 7.2 be numbered 7.2-3,  $\backslash\text{thmcountersep}$  should be  $\backslash\text{def}$ 'ed to ‘-’. Its default is ‘.’.

$\backslash\text{begintheorem}\{\text{NAME}\}\{\text{NUMBER}\}$  : A command that begins a theorem environment for a ‘theorem’ named ‘NAME NUMBER’ – e.g.,  $\backslash\text{begintheorem}\{\text{Lemma}\}\{3.7\}$  starts Lemma 3.7.

$\backslash\text{opargbegintheorem}\{\text{NAME}\}\{\text{NUMBER}\}\{\text{OPARG}\}$  :  
A command that begins a theorem environment for a ‘theorem’ named ‘NAME NUMBER’ with optional argument OPARG – e.g.,  $\backslash\text{begintheorem}\{\text{Lemma}\}\{3.7\}\{\text{Jones}\}$  starts ‘Lemma 3.7 (Jones)’.

$\backslash\text{endtheorem}$  : A command that ends a theorem environment.

$\backslash\text{newtheorem}\{\text{NAME}\}\{\text{TEXT}\}[\text{COUNTER}] ==$

```

BEGIN
 if \NAME is definable
 then \@definecounter{NAME}
 if COUNTER present
 then \@newctr{NAME}[COUNTER] fi
 \theNAME == BEGIN \theCOUNTER \@thmcountersep
 eval\@thmcounter{NAME}
END
 else \theNAME == BEGIN eval\@thmcounter{NAME} END
\NAME == \@thm{NAME}{TEXT}
\endNAME == \@endtheorem
 else error
 fi
END

\newtheorem{NAME}[OLDNAME]{TEXT}==
BEGIN
 if counter OLDNAME nonexistent
 then ERROR
 else
 if \NAME is definable
 then BEGIN
 \theNAME == \theOLDNAME
 \NAME == \@thm{OLDNAME}{TEXT}
 \endNAME == \@endtheorem
 END
 else error
 fi
 fi
END

\@thm{NAME}{TEXT} ==
BEGIN
 \refstepcounter{NAME}
 if next char = [
 then \@ythm{NAME}{TEXT}
 else \@xthm{NAME}{TEXT}
 fi
END

\@xthm{NAME}{TEXT} ==
BEGIN
 \@begintheorem{TEXT}{\theNAME}
 \ignorespaces
END

\@ythm{NAME}{TEXT}[OPARG] ==
BEGIN
 \@opargbegintheorem{TEXT}{\theNAME}{OPARG}
 \ignorespaces

```

END

`\newtheorem` `\newtheorem` ought really be allowed only in the preamble Which would be good document style, and allow some main memory to be saved by declaring these commands to be `\onlypreamble`. Unfortunately the  $\LaTeX$  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 F

## ltsect.dtx

### 59 Sectioning Commands

This file defines the declarations such as `\author` which are used by `\maketitle`. `\maketitle` itself is defined by each class, not in the L<sup>A</sup>T<sub>E</sub>X kernel.

The second part of the file defines the generic commands used for defining sectioning commands such as `\chapter`. Again the actual document level commands are defined in the class files, in terms of these commands.

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

#### 59.1 The Title

```
\title The user defines the title and author by the declarations \title{<name>},
\author \author{<name>}
\date Similarly the date is declared with \date{<date>}.
\thanks Inside these, the \thanks{<footnote text>} command may be used to make
\and acknowledgements, notice of address, etc. in a footnote. If there are multiple
\maketitle authors, they have to be separated with the \and command.
And finally, the \maketitle command produces the actual title, using the
information previously saved with the other commands.

\title \title for use in \maketitle. If not given \maketitle will produce an error
\@title message.
3 \def\title#1{\gdef\@title{#1}}
4 \def\@title{\@latex@error{No \noexpand\title given}\@ehc}

\author \author for use in \maketitle. If not given \maketitle will produce a warning
\@author message.
5 \def\author#1{\gdef\@author{#1}}
6 \def\@author{\@latex@warning@no@line{No \noexpand\author given}}

\date \date for use in \maketitle. If not given \maketitle will produce \today as the
\@date default.
7 \def\date#1{\gdef\@date{#1}}
8 \gdef\@date{\today}

\thanks
9 \def\thanks#1{\footnotemark
10 \protected@xdef\@thanks{\@thanks
11 \protect\footnotetext[\the\c@footnote]{#1}}%
12 }

\@thanks
13 \let\@thanks\@empty
```

```

\and
14 \def\and{% % \begin{tabular}
15 \end{tabular}%
16 \hskip 1em \@plus.17fil%
17 \begin{tabular}[t]{c}}% % \end{tabular}

18 \message{sectioning,}

```

## 59.2 Sectioning

```

\@secpenalty
19 \newcount\@secpenalty
20 \@secpenalty = -300

```

`\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.

```
21 \newif\if@noskipsec \@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=1, section = 2, etc.

**indent:** Indentation of heading from left margin

**beforeskip:** Absolute value = skip to leave above the heading. If negative, then paragraph indent of text following heading is suppressed.

**afterskip:** if positive, then skip to leave below heading, else negative of skip to leave to right of run-in heading.

**style:** Commands to set style. Since June 1996 release the *last* command in this argument may be a command such as `\MakeUppercase` or `\fbox` that takes an argument. The section heading will be supplied as the argument to this command. So setting #6 to, say, `\bfseries\MakeUppercase` would produce bold, uppercase headings.

If ‘\*’ is missing, then increment the counter. If it is present, then there should be no [*altheading*] argument. The command uses the counter ‘secnumdepth’. It contains a pointer to the highest section level that is to be numbered.

**Warning:** The `\@startsection` command should be at the same or higher grouping level as the text that follows it. For example, you should *not* do something like

```

\def\foo{ \begingroup ...
 \paragraph{...}
 \endgroup}

```

Pseudocode for the `\@startsection` command

```
\@startsection
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE} ==
BEGIN
 IF @noskipsec = T THEN \leavevmode FI
 % true if previous section had no body.

 \par
 \@tempskipa := BEFORESKIP
 @afterindent := T
 IF \@tempskipa < 0 THEN \@tempskipa := -\@tempskipa
 @afterindent := F
 FI
 IF @nobreak = true
 THEN \everypar == null
 ELSE \addpenalty{\@secpenalty}
 \advspace{\@tempskipa}
 FI
 IF * next
 THEN \@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
 ELSE \@dblarg{\@sect
 {NAME}{LEVEL}{INDENT}
 {BEFORESKIP}{AFTERSKIP}{STYLE}}
 FI
END

22 \def\@startsection#1#2#3#4#5#6{%
23 \if@noskipsec \leavevmode \fi
24 \par
25 \@tempskipa #4\relax
26 \@afterindenttrue
27 \ifdim \@tempskipa <\z@
28 \@tempskipa -\@tempskipa \@afterindentfalse
29 \fi
30 \if@nobreak
31 \everypar{}%
32 \else
33 \addpenalty\@secpenalty\advspace\@tempskipa
34 \fi
35 \ifstar
36 {\@ssect{#3}{#4}{#5}{#6}}%
37 {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}
```

`\@sect` Pseudocode for the `\@sect` command

```
\@sect{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}[ARG1]{ARG2}
==
BEGIN
 IF LEVEL > \c@secnumdepth
 THEN \@svsec :=L null
 ELSE \refstepcounter{NAME}
 \@svsec :=L BEGIN \@secntformat{#1}\relax END
```

```

FI
IF AFTERSKIP > 0
 THEN \begingroup
 STYLE
 \@hangfrom{\hskip INDENT\@svsec}
 {\interlinepenalty 10000 ARG2\par}
 \endgroup
 \NAMEmark{ARG1}
 \addcontentsline{toc}{NAME}
 { IF LEVEL > \c@secnumdepth
 ELSE \protect\numberline{\theNAME} FI
 ARG1 }
 ELSE \@svsechd == BEGIN STYLE
 \hskip INDENT\@svsec
 ARG2
 \NAMEmark{ARG1}
 \addcontentsline{toc}{NAME}
 { IF LEVEL > \c@secnumdepth
 ELSE
\protect\numberline{\theNAME}
 FI
 ARG1 }
 END
 FI
 \@xsect{AFTERSKIP}
 END
 38 \def\@sect#1#2#3#4#5#6[#7]#8{%
 39 \ifnum #2>\c@secnumdepth
 40 \let\@svsec\@empty
 41 \else
 42 \refstepcounter{#1}%

```

Since \@seccntformat might end with an improper \hskip which is scanning forward for plus or minus we end the definition of \@svsec with \relax as a precaution.

```

 43 \protected@edef\@svsec{\@seccntformat{#1}\relax}%
 44 \fi
 45 \@tempskipa #5\relax
 46 \ifdim \@tempskipa>\z@
 47 \begingroup

```

This { used to be after the argument to \@hangfrom but was moved here to allow commands such as \MakeUppercase to be used at the end of #6.

```

 48 #6{%
 49 \@hangfrom{\hskip #3\relax\@svsec}%
 50 \interlinepenalty \@M #8\@par}%
 51 \endgroup
 52 \csname #1mark\endcsname{#7}%
 53 \addcontentsline{toc}{#1}{%
 54 \ifnum #2>\c@secnumdepth \else
 55 \protect\numberline{\csname the#1\endcsname}%

```

```

56 \fi
57 #7}%
58 \else
\relax added 2 May 90
59 \def\@svsechd{%
60 #6{\hskip #3\relax
61 \@svsec #8}%
62 \csname #1mark\endcsname{#7}%
63 \addcontentsline{toc}{#1}{%
64 \ifnum #2>\c@secnumdepth \else
65 \protect\numberline{\csname the#1\endcsname}%
66 \fi
67 #7}}%
68 \fi
69 \@xsect{#5}}

```

`\@xsect` Pseudocode for the `\@xsect` command

```

\@xsect{AFTERSKIP} ==
BEGIN
 IF AFTERSKIP > 0
 THEN \par \nobreak
 \vskip AFTERSKIP
 \@afterheading
 ELSE @nobreak :=G F
 @noskipsec :=G T
 \everypar{ IF @noskipsec = T
 THEN @noskipsec :=G F
 \clubpenalty :=G 10000
 \hskip -\parindent
 \begingroup
 \@svsechd
 \endgroup
 \unskip
 \hskip -AFTERSKIP \relax
 % relax added 14 Jan 91
 ELSE \clubpenalty :=G \@clubpenalty
 \everypar := NULL
 FI
 }
 FI
 FI
END

```

```

70 \def\@xsect#1{%
71 \@tempskipa #1\relax
72 \ifdim \@tempskipa>\z@

```

Why not combine `\@sect` and `\@xsect` and save doing the same test twice? It is not possible to change this now as these have become hooks!

This `\par` seems unnecessary.

```

73 \par \nobreak
74 \vskip \@tempskipa

```

```

75 \@afterheading
76 \else

77 \@nobreakfalse
78 \global\@noskipsectrue
79 \everypar{%
80 \if@noskipsec
81 \global\@noskipsecfalse
82 {\setbox\z@\lastbox}%
83 \clubpenalty\@M
84 \begingroup \@svsechd \endgroup
85 \unskip
86 \@tempskipa #1\relax
87 \hskip -\@tempskipa
88 \else
89 \clubpenalty \@clubpenalty
90 \everypar{}}%
91 \fi}%
92 \fi
93 \ignorespaces}

```

`\@secntformat` This command formats the section number including the space following it.

```

94 \def\@secntformat#1{\csname the#1\endcsname\quad}

```

Pseudocode for the `\@ssect` command

```

\@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}{ARG} ==
BEGIN
 IF AFTERSKIP > 0
 THEN \begingroup
 STYLE
 \@hangfrom{\hskip INDENT}{\interlinepenalty 10000
ARG\par}
 \endgroup
 ELSE \@svsechd == BEGIN STYLE
 \hskip INDENT
 ARG
 END
 FI
 \@xsect{AFTERSKIP}
END

```

Pseudocode for the `\@afterheading` command

```

\@afterheading ==
BEGIN
 \@nobreak :=G true
 \everypar := BEGIN IF \@nobreak = T
 THEN \@nobreak :=G false
 \clubpenalty :=G 10000
 IF \@afterindent = F
 THEN remove \lastbox
 FI
 ELSE \clubpenalty :=G \@clubpenalty
 \everypar := NULL

```

FI

END

END

`\@ssect`

```
95 \def\@ssect#1#2#3#4#5{%
96 \@tempkipa #3\relax
97 \ifdim \@tempkipa>\z@
98 \begingroup
```

This { used to be after the argument to `\@hangfrom` but was moved here to allow commands such as `\MakeUppercase` to be used at the end of #4.

```
99 #4{%
100 \@hangfrom{\hskip #1}%
101 \interlinepenalty \@M #5\@par}%
102 \endgroup
103 \else
104 \def\@svsechd{#4{\hskip #1\relax #5}}%
105 \fi
106 \@xsect{#3}}
```

`\if@afterindent`

```
\@afterindenttrue 107 \newif\if@afterindent \@afterindenttrue
```

`\@afterheading` This hook is used in setting up custom-built headings in classes.dtx.

```
108 \def\@afterheading{%
109 \@nobreaktrue
110 \everypar{%
111 \if@nobreak
112 \@nobreakfalse
113 \clubpenalty \@M
114 \if@afterindent \else
115 {\setbox\z@\lastbox}%
116 \fi
117 \else
118 \clubpenalty \@clubpenalty
119 \everypar{}}%
120 \fi}}
```

`\@hangfrom` `\@hangfrom{<text>}` : Puts *<text>* in a box, and makes a hanging indentation of the following material up to the first `\par`. Should be used in vertical mode.

```
121 \def\@hangfrom#1{\setbox\@tempboxa\hbox{#1}}%
122 \hangindent \wd\@tempboxa\noindent\box\@tempboxa}
```

`\c@secnumdepth`

```
\c@tocdepth 123 \newcount\c@secnumdepth
124 \newcount\c@tocdepth
```

`\secdef` `\secdef{<unstarcmds>}{<unstarcmds>}{<starcmds>}`

When defining a `\chapter` or `\section` command without using `\@startsection`, you can use `\secdef` as follows:

```
1. \def\chapter{ ... \secdef \<starcmd> \<unstarcmd> }
```



2. `\def\{starcmd}[#1]#2{ ... }` % Command to define `\chapter[...]{...}`

3. `\def\{unstarcmd}#1{ ... }` % Command to define `\chapter*{...}`

```
125 \def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

### 59.2.1 Initializations

```
\sectionmark
\subsectionmark 126 \let\sectionmark@gobble
\subsubsectionmark 127 \let\subsectionmark@gobble
\paragraphmark 128 \let\subsubsectionmark@gobble
\subparagraphmark 129 \let\paragraphmark@gobble
130 \let\subparagraphmark@gobble

131 \message{contents,}
```

## 59.3 Table of Contents etc.

### 59.3.1 Convention

`\tf@{foo}` = file number for output for table foo. The file is opened only if `@filesw = true`.

### 59.3.2 Commands

A `\l@{type}{<entry>}{<page>}` Macro needs to be defined by document style for making an entry of type `<type>` in a table of contents, etc. E.g., the document style should define `\l@chapter`, `\l@section`, etc.

**Note:** When the `\protect` command is used in the `<entry>` or `<text>` of one of the commands below, it causes the following control sequence to be written on the file without being expanded. The sequence will be expanded when the table of contents entry is processed.

**Surprise:** Inside an `\addcontentsline` or `\addtocontents` command argument, the commands: `\index`, `\glossary`, and `\label` are no-ops. This could cause a problem if the user puts an `\index` or `\label` into one of the commands he writes, or into the optional ‘short version’ argument of a `\section` or `\caption` command.

`\@starttoc` The `\@starttoc{<ext>}` command is used to define the commands: `\tableofcontents`, `\listoffigures`, etc.

For example: `\@starttoc{lof}` is used in `\listoffigures`. This command reads the `.<ext>` file and sets up to write the new `.<ext>` file.

```
\@starttoc{EXT} ==
BEGIN
 \begingroup
 \makeatletter
 read file \jobname.EXT
 IF @filesw = true
 THEN open \jobname.EXT as file \tf@EXT
 FI
 @nobreak :=G FALSE %% added 24 May 89
```

```

\endgroup
END

```

```

132 \def\starttoc#1{%
133 \begingroup
134 \makeatletter
135 \@input{\jobname.#1}%
136 \if@filesw
137 \expandafter\newwrite\csname tf@#1\endcsname
138 \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
139 \fi
140 \@nobreakfalse
141 \endgroup}

```

**\addcontentsline** The `\addcontentsline{<table>}{<type>}{<entry>}` command allows the user to add his/her own entry to a table of contents, etc. The command adds the entry `\contentsline{<type>}{<entry>}{<page>}` to the `.<table>` file.

This macro is implemented as an application of `\addtocontents`. Note that `\thepage` is not expandable during `\protected@write` therefore one gets the page number at the time of the `\shipout`.

```

142 \def\addcontentsline#1#2#3{%
143 \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}

```

**\addtocontents** The `\addtocontents{<table>}{<text>}` command adds `<text>` to the `.<table>` file, with no page number.

```

144 \long\def\addtocontents#1#2{%
145 \protected@write\@auxout
146 {\let\label@gobble \let\index@gobble \let\glossary@gobble}%
147 {\string\@writefile{#1}{#2}}}

```

**\contentsline** The `\contentsline{<type>}{<entry>}{<page>}` macro produces a `<type>` entry in a table of contents, etc. It will appear in the `.toc` or other file. For example, The entry for subsection 1.4.3 in the table of contents for example, might be produced by:

```

\contentsline{subsection}
{\makebox{30pt}[r]{1.4.3} Gnats and Gnus}{22}

```

The `\protect` command causes command sequences to be written without expanding them.

```

148 \def\contentsline#1{\csname l@#1\endcsname}

```

`\dottedtocline{<level>}{<indent>}{<numwidth> }{<title>}{<page>}`: Macro to produce a table of contents line with the following parameters:

**level** If `<level> > \c@tocdepth`, then no line produced.

**indent** Total indentation from the left margin.

**numwidth** Width of box for number if the `<title>` has a `\numberline` command. As of 25 Jan 1988, this is also the amount of extra indentation added to second and later lines of a multiple line entry.

**title** Contents of entry.

**page** Page number.

Uses the following parameters, which must be set by the document style. They should be defined with `\def`'s.

**pnumwidth** Width of box in which page number is set.

**tocrmarg** Right margin indentation for all but last line of multiple-line entries.

**dotsep** Separation between dots, in mu units. Should be `\def`'d to a number like 2 or 1.7

`\@dottedtocline`

```
149 \def\@dottedtocline#1#2#3#4#5{%
150 \ifnum #1>\c@tocdepth \else
151 \vskip \z@ \@plus.2\p@
152 {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
153 \parindent #2\relax\@afterindenttrue
154 \interlinepenalty\@M
155 \leavevmode
156 \@tempdima #3\relax
157
158 \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
159 {#4}\nobreak
160 \leaders\hbox{#5}\m@th
```

If a document uses fonts other than computer modern, the use of a dot from math can be very disturbing despite the fact that this might be the only place in a document that then uses computer modern. Therefore we surround the dot with an `\hbox` to escape to the surrounding text font.

```
160 \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
161 mu$}\hfill
162 \nobreak
163 \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
164 \par}%
165 \fi}
```

**Note:** `\nobreak`'s added 7 Jan 86 to prevent bad line break that left the page number dangling by itself at left edge of a new line.

Changed 25 Jan 88 to use `\leftskip` instead of `\hangindent` so leaders of multiple-line contents entries would line up properly.

`\numberline` `\numberline{<number>}`: For use in a `\contentsline` command. It puts `<number>` flushleft in a box of width `\@tempdima` (Before 25 Jan 88 change, it also added `\@tempdima` to the hanging indentation.)

```
166 \def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
167 \endkernel)
```

# File G

## ltfloat.dtx

### 60 Floats

The different types of floats are identified by a  $\langle type \rangle$  name, which is the name of the counter for that kind of float. For example, figures are of type ‘figure’ and tables are of type ‘table’. Each  $\langle type \rangle$  has associated a positive  $\langle type\ number \rangle$ , which is a power of two. E.g., figures might be have type number 1, tables type number 2, programs type number 4, etc.

The locations where a float can go are specified by a  $\langle placement\ specifier \rangle$ , which is a list of the possible locations, each denoted by a letter as follows:

h : here — at the current location in the text.  
t : top — at the top of a text page.  
b : bottom — at the bottom of a text page.  
p : page — on a separate float page

In addition, in conjunction with these, you can use ‘!’ which means that the current values of the float positioning parameters are ignored for this float. (Has no effect on ‘p’, float page positioning.) For example, ‘pht’ specifies that the float can appear in any of three locations: page, here or top.

#### 60.1 Floating Environments

```
1 (*2kernel)
2 \message{floats,}
```

Where floats may appear on a page, and how many may appear there are specified by the following float placement parameters. The numbers are named like counters so the user can set them with the ordinary counter-setting commands.

`\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
 \@capytype ==L TYPE
 \@dblflset
 \@fps ==L PLACEMENT
 \@onelevel@sanitize \@fps
 add default PLACEMENT if at most ! in PLACEMENT ==
\@fpsadddefault
 if inner
 then LaTeX Error: 'Not in outer paragraph mode.'
 \@floatpenalty := 0
 else if \@freelist nonempty
 then \@currbox :=L head of \@freelist
 \@freelist :=G tail of \@freelist
 \@count\@currbox :=G 32*\ftype@TYPE +
 bits determined by
PLACEMENT
 else \@floatpenalty := 0
 LaTeX Error: 'Too many unprocessed floats'
 fi

```

```

fi
\@currbox :=G \color@vbox
\normalcolor
\ vbox{
 %% 15 Dec 87 -
 %% removed \boxmaxdepth :=L 0pt
 %% that made box 0 depth because it screwed
 %% things up. Instead, added \vskip0pt at
end
\hsize = \columnwidth
\@parboxrestore
\@floatboxreset
END

```

```

\caption ==
BEGIN
 \refstepcounter{\@capttype}
 \@dblarg{\caption{\@capttype}}
END

```

In following definition, `\par` moved from after `\addcontentsline` to before `\addcontentsline` because the `\write` could cause an extra blank line to be added to the paragraph above the caption. (Change made 12 Jun 87)

```

\@caption{TYPE}[STEXT]{TEXT} ==
BEGIN
 \par
\addcontentsline{\ext@TYPE}{TYPE}{\numberline{\theTYPE}{STEXT}}
 \begingroup
 \@parboxrestore
 \@normalsize
 \@makecaption{\fnum@TYPE}{TEXT}
 \par
 \endgroup
END

```

`\@dblfloat{TYPE}[PLACEMENT]` : Macro to begin a float environment for

a double-column float of type TYPE with PLACEMENT as the placement

specifier. The default value of PLACEMENT is 'tp'

The environment is ended by `\end@dblfloat`.

E.g., `\figure* == \@dblfloat{figure}`,

`\endfigure* == \end@dblfloat`.

```

\@dblfloat{TYPE}[PLACEMENT] ==

```

Identical to `\@float{TYPE}[PLACEMENT]` except `\hsize` and `\linewidth` are set to `\textwidth`.

`\@floatpenalty`

```
3 \newcount\@floatpenalty
```

`\caption` This is set to be an error message outside a float since no `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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates  $\text{T}_{\text{E}}\text{X}$ 's 'top of page' behaviour. The flag must be set false at the start of the first paragraph. This is achieved by a redefinition of `\everypar`, but the call to `\@parboxrestore` removes that redefinition, so it is re-inserted if needed. If the flag is already false then the `\caption` was not the first entry in the float, and so some other paragraph has already activated the special `\everypar`. In this case no further action is needed.

```
19 \@parboxrestore
20 \if@minipage
21 \@setminipage
22 \fi
23 \normalsize
24 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
25 \endgroup}
```

`\@float`

`\@dblflset`

```
26 \def\@float#1{%
27 \@ifnextchar[%
28 {\@xfloat{#1}}%
29 {\edef\reserved@a{\noexpand\@xfloat{#1}[\csname fps@#1\endcsname]}}%
30 \reserved@a}}
```

`\@dblfloat`

```
31 \def\@dblfloat{%
32 \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
33 \reserved@a}
```

`\fps@dbl` Note that all double floats have default fps ‘tp’.

`\@setfps` This sets the fps, dealing with error conditions by adding the default.

`\@xfloat` The first part of this sets the count register that stores all the information about the type and fps of the float.

We assume here that the default specifiers already contain no active characters. It may be better to store the defaults as numbers, rather than symbol strings.

```
34 </2kernel>
35 <latexrelease>\IncludeInRelease{2015/01/01}%
36 <latexrelease> \{\@xfloat}{Check float options}%
37 <*2kernel | latexrelease>
38 \def\@xfloat #1[#2]{%
39 \nodocument
40 \def \@capttype {#1}%
41 \def \@fps {#2}%
42 \@onelevel@sanitize \@fps
43 \def \reserved@b {!}%
44 \ifx \reserved@b \@fps
45 \@fpsadddefault
46 \else
47 \ifx \@fps \@empty
48 \@fpsadddefault
49 \fi
50 \fi
51 \ifhmode
52 \@bsphack
53 \@floatpenalty -\@Mii
54 \else
55 \@floatpenalty-\@Miii
56 \fi
57 \ifinner
58 \@parmoderr\@floatpenalty\z@
59 \else
60 \@next\@currbox\@freelist
61 {%
62 \@tempcnta \sixt@n
63 \expandafter \@tfor \expandafter \reserved@a
64 \expandafter :\expandafter =\@fps
65 \do
```

Start of changes, use a nested if structure, ending in an error.

```
66 {%
67 \if \reserved@a h%
68 \ifodd \@tempcnta
69 \else
70 \advance \@tempcnta \@ne
71 \fi
72 \else\if \reserved@a t%
```



```

73 \@setfpsbit \tw@
74 \else\if \reserved@a b%
75 \@setfpsbit 4%
76 \else\if \reserved@a p%
77 \@setfpsbit 8%
78 \else\if \reserved@a !%
79 \ifnum \@tempcnta>15
80 \advance\@tempcnta -\sixt@n\relax
81 \fi
82 \else
83 \@latex@error{Unknown float option '\reserved@a'}%
84 {Option '\reserved@a' ignored and 'p' used.}%
85 \@setfpsbit 8%
86 \fi\fi\fi\fi\fi
87 }%

```

End of changes

```

88 \@tempcntb \csname ftype@\@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@
```

We make sure that we never exceed `\textheight`, otherwise float will never get typeset (91/03/15 FMi).

```
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 \ifnum\@floatpenalty <-\@Mii \@Esphack\fi
239 \else
240 \end@float
241 \fi
242 \fi
243 \fi
244 \fi
245 }%
246 <latexrelease>\EndIncludeInRelease
247 <*2kernel>
```

Force the depth of two column float boxes.

What follows is essentially `\end@float` without a starting `\@endfloatbox`.

We make sure that we never exceed `\textheight`, otherwise float will never get typeset (91/03/15 FMi).

RmS 92/03/18 changed `\@esphack` to `\@Esphack`.

`\@endfloatbox` This macro is not intended to be a hook; it is designed to help maintain the integrity of this code, which is used twice and, as can be seen, is subject to frequent changes.

```

248 \def \@endfloatbox{%
249 \par\vskip\z@skip %% \par\vskip\z@ added 15 Dec 87

250 \@minipagefalse
251 \outer@nobreak
252 \egroup %% end of vbox
253 \color@endbox
254 }
255 %
256 % \begin{macro}{\outer@nobreak}
257 % \changes{v1.0h}{1994/05/20}{Macro added: default is to do nothing.}
258 % \begin{macrocode}
259 \let\outer@nobreak\@empty

```

`\@largefloatcheck` This calculates by how much a float is oversize for the page and prints this in a warning message.

```

260 \def \@largefloatcheck{%
261 \ifdim \ht\@currbox>\textheight
262 \@tempdima -\textheight
263 \advance \@tempdima \ht\@currbox

264 \@latex@warning {Float too large for page by \the\@tempdima}%
265 \ht\@currbox \textheight
266 \fi
267 }

```

`\@dbflt`

```

\@xdblfloat 268 \def\@dbflt#1{\ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
269 \def\@xdblfloat#1[#2]{%
270 \xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}

```

Moved to ltoutput 93/12/16

```

271 %\newcount\c@topnumber
272 %\newcount\c@dbltopnumber
273 %\newcount\c@bottomnumber
274 %\newcount\c@totalnumber

```

`\@dblfloatplacement` An analysis of `\@floatplacement`:  
This should be called whenever `\@colht` has been set.

```

275 \def\@floatplacement{\global\@topnum\c@topnumber
276 % Textpage bit, global:
277 \global\@toproom \topfraction\@colht
278 \global\@botnum \c@bottomnumber
279 \global\@botroom \bottomfraction\@colht
280 \global\@colnum \c@totalnumber
281 % Floatpage bit, local:
282 \@fpmin \floatpagefraction\@colht}
283 (/2kernel)

```

`\@dblfloatplacement` This should be called only within a group. Now changed to provide extra checks in `\@addtodblcol`, needed when processing a BANG float.

```
284 (latexrelease)\IncludeInRelease{2015/01/01}%
285 (latexrelease) {\@dblfloatplacement}{float order in 2-column}%
286 (*2ekernel | latexrelease)
```

When making two column float area, look for floats with 1sp depth.

```
287 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
288 \global\@dbltoproom \dbltopfraction\@colht
289 \@textmin \@colht
290 \advance \@textmin -\@dbltoproom
291 \@fpmin \dblfloatpagefraction\textheight
292 \@fptop \@dblfpptop
293 \@fpsep \@dblfpsep
294 \@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, `\@dblfloatplacement` needs to be called inside a group which is a questionable design.

```
295 \def\f@depth{1sp}}%
296 (/2ekernel | latexrelease)
297 (latexrelease)\EndIncludeInRelease
298 (latexrelease)\IncludeInRelease{0000/00/00}%
299 (latexrelease) {\@dblfloatplacement}{float order in 2-column}%
300 (latexrelease)\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
301 (latexrelease) \global \@dbltopnum \c@dbltopnumber
302 (latexrelease) \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses `\@textmin` to locally store the amount of extra room in the column.

```
303 (latexrelease) \@textmin \@colht
304 (latexrelease) \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
305 (latexrelease) \@fpmin \dblfloatpagefraction\textheight
306 (latexrelease) \@fptop \@dblfpptop
307 (latexrelease) \@fpsep \@dblfpsep
308 (latexrelease) \@fpbot \@dblfpbot
309 (latexrelease)}%
310 (latexrelease)\EndIncludeInRelease
311 (*2ekernel)
```

#### 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

```

\marginpar

```

312 \def\marginpar{%
313 \ifhmode
314 \@bsphack
315 \@floatpenalty -\@Mii
316 \else
317 \@floatpenalty-\@Miii
318 \fi
319 \ifinner
320 \@parmoderr
321 \@floatpenalty\z@
322 \else
323 \@next\@currbox\@freelist{}{}%
324 \@next\@marbox\@freelist{\global\count\@marbox\m@ne}%
325 {\@floatpenalty\z@
326 \@fltovf\def\@currbox{\@tempboxa}\def\@marbox{\@tempboxa}}%
327 \fi
328 \@ifnextchar [\@xmpar\@ympar}

```

\@xmpar

```

329 \long\def\@xmpar[#1]#2{%
330 \@savemarbox\@marbox{#1}%
331 \@savemarbox\@currbox{#2}%
332 \@xympar}

```

\@ympar

```

333 \long\def\@ympar#1{%
334 \@savemarbox\@marbox{#1}%
335 \global\setbox\@currbox\copy\@marbox
336 \@xympar}

```

\@savemarbox

```

337 \long\def \@savemarbox #1#2{%
338 \global\setbox #1%
339 \color@vbox
340 \vtop{%
341 \hsize\marginparwidth
342 \@parboxrestore
343 \@marginparreset
344 #2%
345 \@minipagefalse
346 \outer@nobreak
347 }%
348 \color@endbox
349 }

```

\@marginparreset The rationale for allowing these normally global flags to be set locally here, via \@parboxrestore was stated originally by Donald Arsenau and extended by Chris



Rowley. It is because these flags are only set globally to true by section commands, and these should never appear within marginals or floats or, indeed, in any group; and they are only ever set globally to false when they are definitely true.

If anyone is unhappy with this argument then both flags should be treated as in `\set@nobreak`; otherwise this command will be redundant.

```

350 \def \@marginparreset {%
351 \reset@font
352 \normalsize
353 % \let@if@nobreak\iffalse
354 % \let@if@noskipsec\iffalse
355 % \setnobreak
356 \@setminipage
357 }

```

`\@xympar`

Setting the box here is done only because the code uses `\end@float`; it will be empty and gets discarded.

```

358 \def \@xympar{%
359 \ifnum@floatpenalty <z@\@cons\@currlist\@marbox\fi
360 \setbox\@tempboxa
361 \color@vbox
362 \vbox \bgroup
363 \end@float
364 \@ignorefalse
365 \@esphack
366 }

```

`\reversemarginpar`

`\normalmarginpar`

```

367 \def\reversemarginpar{\global\@mparbottom\z@ \@reversemargintrue}
368 \def\normalmarginpar{\global\@mparbottom\z@ \@reversemarginfalse}
369 \message{footnotes,}

```

## 60.2 Footnotes

`\footnote{NOTE}` : User command to insert a footnote.

`\footnote[NUM]{NOTE}`: User command to insert a footnote numbered *NUM*, where *NUM* is a number – 1, 2, etc. For example, if footnotes are numbered \*, \*\*, etc. within pages, then `\footnote[2]{...}` produces footnote '\*\*'. This command does not step the footnote counter.

`\footnotemark[NUM]` : Command to produce just the footnote mark in the text, but no footnote. With no argument, it steps the footnote counter before generating the mark.

`\footnotetext[NUM]{TEXT}` : Command to produce the footnote but no mark. `\footnote` is equivalent to

`\footnotemark \footnotetext .`

As in PLAIN, footnotes use `\insert\footins`, and the following parameters:

- `\footnotesize` : Size-changing command for footnotes.
- `\footnotesep` : The height of a strut placed at the beginning of every footnote.
- `\skip\footins` : Space between main text and footnotes. The rule separating footnotes from text occurs in this space. This space lies above the strut of height `\footnotesep` which is at the beginning of the first footnote.
- `\footnoterule` : Macro to draw the rule separating footnotes from text. It is executed right after a `\vspace` of `\skip\footins`. It should take zero vertical space—i.e., it should to a negative skip to compensate for any positive space it occupies. (See PLAIN.TEX.)
- `\interfootnotelinepenalty` : Interline penalty for footnotes.
- `\thefootnote` : In usual LaTeX style, produces the footnote number. If footnotes are to be numbered within pages, then the document style file must include an `\@addtoreset` command to cause the footnote counter to be reset when the page counter is stepped. This is not a good idea, though, because the counter will not always be reset in time to ensure that the first footnote on a page is footnote number one.
- `\@thefnmark` : Holds the current footnote's mark—e.g., `\dag` or `'1'` or `'a'`.
- `\@mpfnnumber` : A macro that generates the numbers for `\footnote` and `\footnotemark` commands. It `== \thefootnote` outside a minipage environment, but can be changed inside to generate numbers for `\footnote`'s.
- `\@makefnmark` : A macro to generate the footnote marker from `\@thefnmark`. The default definition was `\hbox{${\@thefnmark}}`.
- This is now replaced by `\textsuperscript{\@thefnmark}`
- `\@makefntext{NOTE}` :  
Must produce the actual footnote, using `\@thefnmark` as the mark

of the footnote and NOTE as the text. It is called when effectively inside a `\parbox`, with `\hsize = \columnwidth`.

For example, it might be as simple as

```
$^{\@thefnmark}$ NOTE
```

In a minipage environment, `\footnote` and `\footnotetext` are redefined so that

- (a) they use the counter `mpfootnote`
- (b) the footnotes they produce go at the bottom of the minipage.

The switch is accomplished by letting `\@mpfn == footnote` or `mpfootnote` and `\thempfn == \thefootnote` or `\thempfootnote`, and by redefining `\@footnotetext` to be `\@mpfootnotetext` in the minipage.

```
\footnote{NOTE} ==
BEGIN
 \stepcounter{\@mpfn}
 begingroup
 \protect == \noexpand
 \@thefnmark :=G eval(\thempfn)
 endgroup
 \@footnotemark
 \@footnotetext{NOTE}
END
```

```
\footnote[NUM]{NOTE} ==
BEGIN
 begingroup
 \protect == \noexpand
 counter \@mpfn :=L NUM
 \@thefnmark :=G eval(\thempfn)
 endgroup
 \@footnotemark
 \@footnotetext{NOTE}
END
```

```
\footnotemark ==
BEGIN \stepcounter{footnote}
 begingroup
 \protect == \noexpand
 \@thefnmark :=G eval(\thefootnote)
 endgroup
 \@footnotemark
END
```

```
\footnotemark[NUM] ==
BEGIN
 begingroup
 footnote counter :=L NUM
 \protect == \noexpand
 \@thefnmark :=G eval(\thefootnote)
 endgroup
END
```

```

 endgroup
 \@footnotemark
 END

\@footnotemark ==
 BEGIN
 \leavevmode
 IF hmode THEN \@x@sf := \the\spacefactor FI
 \@makefnmark % put number in main text
 IF hmode THEN \spacefactor := \@x@sf FI
 END

\footnotetext ==
 BEGIN begingroup \protect == \noexpand
 \@thefnmark :=G eval (\thempfn)
 endgroup
 \@footnotetext
 END

\footnotetext[NUM] ==
 BEGIN begingroup counter \@mpfn :=L NUM
 \protect == \noexpand
 \@thefnmark :=G eval (\thempfn)
 endgroup
 \@footnotetext
 END

\footins LATEX does use the same insert for footnotes as PLAIN.
370 \newinsert\footins
 LATEX leaves these initializations for the \footins insert.
371 \skip\footins=\bigskipamount % space added when footnote is present
372 \count\footins=1000 % footnote magnification factor (1 to 1)
373 \dimen\footins=8in % maximum footnotes per page

\footnoterule LATEX keeps PLAIN TEX's \footnoterule as the default.
374 \def\footnoterule{\kern-3\p@
375 \hrule \width 2in \kern 2.6\p@} % the \hrule is .4pt high

\thefootnote
376 \@definecounter{footnote}
377 \def\thefootnote{\@arabic\c@footnote}

\thempfootnote The default display for the footnote counter in minipages is to use italic letters.
 We use \itshape not \textit as the latter would add an italic correction.
378 \@definecounter{mpfootnote}
379 \def\thempfootnote{{\itshape\@alph\c@mpfootnote}}

\@makefnmark Default definition.
380 %\def\@makefnmark{\hbox{\$^{\@thefnmark}\m@th}}
381 \def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}

```

`\textsuperscript` This command provides superscript characters in the current text font. It's implementation might change!!!

```
382 \DeclareRobustCommand*\textsuperscript[1]{%
383 \@textsuperscript{\selectfont#1}}
```

`@textsuperscript` This command should not be used directly, but may be used to define other commands `\textsuperscript`, `\makefnmark`. #1 should always start with a font selection command, to activate the font size switch.

```
384 \def\@textsuperscript#1{%
385 {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}
```

`\textsubscript`

```
386 (/2ekernel)
387 (latexrelease)\IncludeInRelease{2015/01/01}%
388 (latexrelease) {\textsubscript}{\textsubscript}%
389 (*2ekernel | latexrelease)

390 \DeclareRobustCommand*\textsubscript[1]{%
391 \@textsubscript{\selectfont#1}}%
```

`@textsubscript`

```
392 \def\@textsubscript#1{%
393 {\m@th\ensuremath_{\mbox{\fontsize\sf@size\z@#1}}}}%

394 (/2ekernel | latexrelease)
395 (latexrelease)\EndIncludeInRelease
396 (latexrelease)\IncludeInRelease{0000/00/00}%
397 (latexrelease) {\textsubscript}{\textsubscript}%
398 (latexrelease)\let\textsubscript\undefined
399 (latexrelease)\let\@textsubscript\undefined
400 (latexrelease)\EndIncludeInRelease
401 (*2ekernel)

402 \def\@textsubscript#1{%
403 {\m@th\ensuremath_{\mbox{\fontsize\sf@size\z@#1}}}}
```

`\footnotesep`

```
404 \newdimen\footnotesep
```

`\footnote`

```
405 \def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
406 \protected@xdef\@thefnmark{\thempfn}%
407 \@footnotemark\@footnotetext}}
```

`\@xfootnote`

```
408 \def\@xfootnote[#1]{%
409 \begingroup
410 \csname c@\@mpfn\endcsname #1\relax
411 \unrestored@protected@xdef\@thefnmark{\thempfn}%
412 \endgroup
413 \@footnotemark\@footnotetext}
```

```

\@footnotetext
414 \long\def\@footnotetext#1{\insert\footins{%
415 \reset@font\footnotesize
416 \interlinepenalty\interfootnotelinepenalty
417 \splittopskip\footnotesepp
418 \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
419 \hsize\columnwidth \@parboxrestore
420 \protected@edef\@currentlabel{%
421 \csname p@footnote\endcsname\@thefnmark
422 }%
423 \color@begingroup
424 \@makefnmark{%
425 \rule{z@\footnotesepp\ignorespaces#1\@finalstrut\strutbox}%
426 \color@endgroup}}%

\footnotemark
427 \def\footnotemark{%
428 \@ifnextchar [\xfootnotemark
429 {\stepcounter{footnote}%
430 \protected@xdef\@thefnmark{\thefootnote}%
431 \@footnotemark}}

\xfootnotemark
432 \def\xfootnotemark[#1]{%
433 \begingroup
434 \c@footnote #1\relax
435 \unrestored@protected@xdef\@thefnmark{\thefootnote}%
436 \endgroup
437 \@footnotemark}

\@footnotemark
438 \def\@footnotemark{%
439 \leavevmode
440 \ifhmode\edef\x@sf{\the\spacefactor}\nobreak\fi
441 \@makefnmark
442 \ifhmode\spacefactor\x@sf\fi
443 \relax}

\footnotetext
444 \def\footnotetext{%
445 \@ifnextchar [\xfootnotenext
446 {\protected@xdef\@thefnmark{\thempfn}%
447 \@footnotetext}}

\xfootnotenext
448 \def\xfootnotenext[#1]{%
449 \begingroup
450 \csname c@\mpfn\endcsname #1\relax
451 \unrestored@protected@xdef\@thefnmark{\thempfn}%
452 \endgroup
453 \@footnotetext}

```

```
\thempfn
 \@mpfn 454 \def\@mpfn{footnote}
 455 \def\thempfn{\thefootnote}
 456 \endkernel
```

# File H

## ltidxglo.dtx

### 61 Index and Glossary Generation

Index and Glossary commands.

```
\makeindex A preamble command to turn on indexing.
\makeglossary A preamble command to turn on making glossary entries.
 \index Make an index entry for #1.
 \glossary Make a glossary entry for #1.
\makeindex ==
 BEGIN
 \index == BEGIN \@bsphack
 \begingroup
 \protect{X} == \string X\space
 %% added 3 Feb 87 for \index
commands
 %% in \footnotes
 re-\catcode special characters
 to 'other'
 \@wrindex
 END

 \@wrindex{ITEM} ==
 BEGIN
 write of {\indexentry{ITEM}{page number}}
 \endgroup
 \@esphack
 END

INITIALIZATION:

\index == BEGIN \@bsphack
 \begingroup
 re-\catcode special characters (in case '%' there)
 \@index
 END

\@index{ITEM} == BEGIN \endgroup \@esphack END

Changes made 14 Apr 89 to write \glossaryentry's instead of
\indexentry's on the .glo file.
1 (*2kernel)
2 \message{index,}

\makeindex
3 \def\makeindex{%
4 \newwrite\@indexfile
```



```

5 \immediate\openout\@indexfile=\jobname.idx
6 \def\index{\@bsphack\beginingroup
7 \@sanitize
8 \@wrindex}\typeout
9 {Writing index file \jobname.idx}%

```

Opening the write channel should be done only once since on some OS multiple opens are forbidden and in any case it is useless. So we turn this into a no-op after use.

```

10 \let\makeindex\@empty
11 }
12 \@onlypreamble\makeindex

```

\@wrindex

```

13 \def\@wrindex#1{%
14 \protected@write\@indexfile{%
15 {\string\indexentry{#1}{\thepage}}%
16 \endgroup
17 \@esphack}

```

\index

```

18 \def\index{\@bsphack\beginingroup \@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\beginingroup
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\beginingroup\@sanitize\@index}
36 </2kernel)

```

# File I

## ltbibl.dtx

### 62 Bibliography Generation

A bibliography is created by the `thebibliography` environment, which generates a title such as “References”, and a list of entries. The `BIBTEX` program will create a file containing such an environment, which will be read in by the `\bibliography` command. With `BIBTEX`, the following commands will be used.

`\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,}
```

#### 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`.

```

\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 {\@citeofmt{\csname b@\@citeb\endcsname}}{#1}}

```

```

\bibdata
\bibstyle 25 \let\bibdata=\@gobble
 26 \let\bibstyle=\@gobble

```

```

\bibliography
 27 \def\bibliography#1{%
 28 \if@filesw
 29 \immediate\write\@auxout{\string\bibdata{#1}}%
 30 \fi
 31 \@input@{\jobname.bbl}}

```

```

\bibliographystyle
 32 \def\bibliographystyle#1{%
 33 \ifx\@begindocumenthook\@undefined\else
 34 \expandafter\AtBeginDocument
 35 \fi
 36 {\if@filesw
 37 \immediate\write\@auxout{\string\bibstyle{#1}}%
 38 \fi}}

```

`\nocite` (Added 14 Jun 85)

This puts information on the `.aux` file that causes `LATEX` 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 `BIBTEX` in the first place).

```
51 \xandafter\let\csname b@*\endcsname\@empty
```

## 62.1 Default definitions

This hook determines the ‘relative formatting’ of the two logical parts of a citation with comment.

`\@cite`

```
52 \def\@cite#1#2{[#1\if@tempswa , #2\fi]}
```

`\@cite@ofmt`

This is, in general, a command that appears to have one argument whose value is, in the kernel, a single cs whose name is the expansion of `b@\@citeb`; the expansion of this cs will typically be some hmode material that produces the detailed typeset form of just the citations themselves.

```
53 \let\@cite@ofmt\hbox
```

`\@biblabel`

```
54 \def\@biblabel#1{[#1]}
```

```
55 </2ekernel>
```

# File J

## ltpage.dtx

### 63 Page styles and related commands

#### 63.1 Page Style Commands

`\pagestyle{<style>}` : sets the page style of the current and succeeding pages to *style*

`\thispagestyle{<style>}` : sets the page style of the current page only to *style*.

To define a page style *style*, you must define `\ps@style` to set the page style parameters.

#### 63.2 How a page style makes running heads and feet

The `\ps@...` command defines the macros `\@oddhead`, `\@oddfoot`, `\@evenhead`, and `\@evenfoot` to define the running heads and feet. (See output routine.) To make headings determined by the sectioning commands, the page style defines the commands `\chaptermark`, `\sectionmark`, etc., where `\chaptermark{<text>}` is called by `\chapter` to set a mark. The `\...mark` commands and the `\...head` macros are defined with the help of the following macros.

(All the `\...mark` commands should be initialized to no-ops.)

#### 63.3 marking conventions

L<sup>A</sup>T<sub>E</sub>X extends T<sub>E</sub>X's `\mark` facility by producing two kinds of marks a 'left' and a 'right' mark, using the following commands:

`\markboth{<left>}{<right>}` : Adds both marks.

`\markright{<right>}` : Adds a 'right' mark.

`\leftmark` : Used in the output routine, gets the current 'left' mark. Works like T<sub>E</sub>X's `\botmark`.

`\rightmark` : Used in the output routine, gets the current 'right' mark. Works like T<sub>E</sub>X's `\firstmark`. The marking commands work reasonably well for right marks 'numbered within' left marks—e.g., the left mark is changed by a `\chapter` command and the right mark is changed by a `\section` command. However, it does produce somewhat anomalous results if 2 `\markboth`'s occur on the same page.

Commands like `\tableofcontents` that should set the marks in some page styles use a `\@mkboth` command, which is `\let` by the `pagestyle` command (`\ps@...`) to `\markboth` for setting the heading or to `\@gobbletwo` to do nothing.

```
1 (*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

```

`\markboth` User commands for setting L<sup>A</sup>T<sub>E</sub>X marks.

`\markright` Test for `\@nobreak` added 15 Apr 86 in `\markboth` and `\markright` letting `\label` and `\index` to `\relax` added 22 Feb 86 so these commands can appear in sectioning command arguments RmS 91/06/21 Same for `\glossary`

```

18 \def\markboth#1#2{%
19 \begingroup
20 \let\label\relax \let\index\relax \let\glossary\relax
21 \unrestored@protected@xdef\@themark {#{1}#{2}}%
22 \@temptokena \expandafter{\@themark}%
23 \mark{\the\@temptokena}%
24 \endgroup
25 \if@nobreak\ifvmode\nobreak\fi\fi}
26 \def\markright#1{%
27 \begingroup
28 \let\label\relax \let\index\relax \let\glossary\relax

```

Protection is handled inside `\@markright`.

```

29 \expandafter\@markright\@themark {#1}%
30 \@temptokena \expandafter{\@themark}%
31 \mark{\the\@temptokena}%
32 \endgroup
33 \if@nobreak\ifvmode\nobreak\fi\fi}

```

`\@markright`

```

\leftmark 34 \def\@markright#1#2#3{\@temptokena {#1}%
\rightmark 35 \unrestored@protected@xdef\@themark{\the\@temptokena}{#3}}
36 \def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
37 \def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}

```

`\@themark` Initialise L<sup>A</sup>T<sub>E</sub>X's marks without setting a T<sub>E</sub>X mark (*whatsit*).

```

38 \def\@themark{}{}

```

`\mark` Test versions of L<sup>A</sup>T<sub>ε</sub>X initialised T<sub>E</sub>X's `\mark` system at this point, but this was removed before the first release.

```
\AtBeginDocument{\mark{}}}
```

`\raggedbottom` `\raggedbottom` typesets pages with no vertical stretch, so they have their natural height instead of all being exactly the same height. (Uses a space of .0001fil to avoid interfering with the 1fil space of `\newpage`.)

```
39 \def\raggedbottom{%
40 \def\@textbottom{\vskip \z@ \@plus.0001fil}\let\@texttop\relax}
```

`\flushbottom` `\flushbottom`: Inverse of `\raggedbottom` — makes all pages the same height.

```
41 \def\flushbottom{%
42 \let\@textbottom\relax \let\@texttop\relax}
```

`\sloppy` `\sloppy` will never (well, hardly ever) produce overfull boxes, but may produce underfull ones. (14 June 85)

```
43 \def\sloppy{%
44 \tolerance 9999%
45 \emergencystretch 3em%
46 \hfuzz .5\p@
47 \vfuzz\hfuzz}
```

`sloppy` A `sloppy` environment is equivalent to `{\par \sloppy ... \par}`.

```
48 \def\sloppy{\par\sloppy}
49 \def\endsloppy{\par}
```

`\fussy` Resets T<sub>E</sub>X's parameters to their normal finicky values.

```
50 \def\fussy{%
51 \emergencystretch\z@
52 \tolerance 200%
53 \hfuzz .1\p@
54 \vfuzz\hfuzz}
```

`\overfullrule` L<sup>A</sup>T<sub>ε</sub>X default is no overfull box rule. Changed by document class option.

```
55 \overfullrule 0pt

56 </2kernel>
```



# File K

## ltoutput.dtx

### 64 Output Routine

#### 64.1 Floats

The ‘2ekernel’ code ensures that a `\usepackage{autoout1}` is essentially ignored if a ‘full’ format is being used that has the autoloading file mode already in the format.

```
1 (defx)\begingroup
2 (defx)\makeatletter
3 (defx)\nfss@catcodes
4 (2ekernel)\expandafter\let\csname ver@autoout1.sty\endcsname\fmtversion
5 (*2ekernel)
6 \message{output,}

 * 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

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><br><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 `\LIST` having the same type as float I, you run `\@bitor` with  $\backslash\text{NUM} = [(\backslash\text{count } I)/32] * 32$ .

```
\@bitor\NUM\LIST ==
BEGIN
 @test :=G false
 { \@elt \CTR == if \NUM <> 0 then
 if \count\CTR / \NUM is odd
 then @test := true fi fi
 \LIST
 }
END
```

`\@cons\LIST\NUM` : Globally sets `\LIST := \LIST * \@elt \NUM`

```
\@cons\LIST\NUM ==
BEGIN { \@elt == \relax
 \LIST :=G \LIST \@elt \NUM
 }
```

#### BOX LISTS FOR FLOAT-PLACEMENT ALGORITHMS

`\@freelist` : List of empty boxes for placing new floats.  
`\@toplist` : List of floats to go at top of current column.  
`\@midlist` : List of floats in middle of current column.  
`\@botlist` : List of floats to go at bottom of current column.  
`\@deferlist` : List of floats to go after current column.  
`\@dbltoplist` : List of double-col. floats to go at top of current page.  
`\@dbldeferlist` : List of double-column floats to go on subsequent pages.

#### FLOAT-PLACEMENT ALGORITHMS

`\@addtobot` : Tries to put insert `\@currbox` on `\@botlist`.

Called only when:

- \* `\ht BOX < \@colroom`
- \* type of `\@currbox` not on `\@deferlist`
- \* `\@colnum > 0`
- \* `@insert = false`

If it succeeds, then:

- \* sets `@insert true`
- \* decrements `\@botroom` by `\ht BOX`
- \* decrements `\@botnum` and `\@colnum` by 1

```

* decrements \@colroom by \ht BOX + either
\floatsep
 or \textfloatsep, as appropriate.
* sets \maxdepth to 0pt

\@addtotoporbot : Tries to put insert \@currbox on \@toplist or
\@botlist.
Called only under same conditions as \@addtobot.
If it succeeds, then:
 * sets @insert true
 * decrements \@toproom or \@botroom by \ht
BOX
 * decrements \@colnum and either \@topnum or
\@botnum by 1
 * decrements \@colroom by \ht BOX +
\floatsep
 or \textfloatsep, as appropriate.

\@addtocurcol : Tries to add \@currbox to current column, setting
@insert true if it succeeds, false otherwise.
It will add \@currbox to top only if bit 0 of
\count \@currbox is 0, and to the bottom only if
bit 0 = 0 or an earlier float of the same type is
put on the bottom.
If the float is put in the text, then
\penalty\interlinepenalty is put
right after the float, before the following \vskip,
and \outputpenalty :=L 0.

\@addtonextcol : Tries to add \@currbox to the next column, setting
@insert true if it succeeds, false otherwise.

\@addtodblcol : Tries to add \@currbox to the next double-column page,
adding it to \@dbltoplist if it succeeds and
\@dbldeferlist if it fails.

\@addmarginpar ==
BEGIN
 if \@currlist nonempty
 then remove \@marbox from \@currlist
 add \@marbox and \@currbox to \@freelist
 %% NOTE: \@currbox = left box
 else LaTeX error: ? %% shouldn't happen
 fi
 \@tempcnta := 1 %% 1 = right, -1 = left
 if @twocolumn = true
 then if @firstcolumn = true
 then \@tempcnta := -1
 fi

```

```

else if @mparswitch = true
 then if count0 odd
 else \@tempcnta := -1
 fi
 fi
 if @reversemargin = true
 then \@tempcnta := -\@tempcnta
 fi
 fi
if \@tempcnta < 0 then \box\@marbox :=G \box\@currbox
fi
\@tempdima :=L maximum(\@mparbottom - \@pageht
 + ht of \@marbox, 0)
if \@tempdima > 0 then LaTeX warning: 'marginpar moved' fi
\@mparbottom :=G \@pageht + \@tempdima + depth of \@marbox
 + \marginparpush
\@tempdima :=L \@tempdima - ht of \@marbox
\box\@marbox :=G \box\@currbox
 \vbox { \vskip \@tempdima
 \box\@marbox
 }
height of \@marbox :=G depth of \@marbox :=G 0
\kern -\@pagedp
\nointerlineskip
\hbox{ if @tempcnta > 0 then \hskip \columnwidth
 \hskip \marginparsep
 else \hskip -\marginparsep
 \hskip -\marginparwidth
 fi
 \box\@marbox \hss
 }
\nobreak
\nointerlineskip
\hbox{\vrule height 0 width 0 depth \@pagedp}
END

```

Floats and marginpars add a lot of dead cycles.

```

7 \maxdeadcycles = 100

8 \let\elt\relax

9 \def\@next#1#2#3#4{\ifx#2\@empty #4\else
10 \expandafter\@xnext #2\@#1#2#3\fi}

11 \def\@xnext \@elt #1#2\@#3#4{\def#3{#1}\gdef#4{#2}}
\changes{v1.1v}{1996/07/26}{put \cs{global} into definition}

12 \def\@testfalse{\global\let@if@test\iffalse}
13 \def\@testtrue {\global\let@if@test\iftrue}
14 \@testfalse

```

\changes{v1.1v}{1996/07/26}{remove \cs{global} before \cs{@test...}}

```
15 \def\@bitor#1#2{\@testfalse {\let\@elt\@xbitor
16 \@tempcnta #1\relax #2}}
```

RmS 91/11/22: Added test for |\count#1 = 0|.  
Suggested by Chris Rowley.

\changes{v1.1v}{1996/07/26}{remove \cs{global} before \cs{@test...}}

```
17 \def\@xbitor #1{\@tempcntb \count#1
18 \ifnum \@tempcnta =\z@
19 \else
20 \divide\@tempcntb\@tempcnta
21 \ifodd\@tempcntb \@testtrue\fi
22 \fi}
```

DEFINITION OF FLOAT BOXES:

\changes{v1.3a}{2015/09/205}  
{extended \cs{@freelist}}

```
23 \kernel)
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 \kernel)
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 \kernel | 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 (*2ekernel)

64 \gdef\@toplist{}
65 \gdef\@botlist{}
66 \gdef\@midlist{}
67 \gdef\@currlist{}
68 \gdef\@deferlist{}
69 \gdef\@dbltoplist{}
70 % \begin{macrocode}
71 % \changes{v1.2m}{2015/03/12}
72 % {initialise \cs{@dbldeferlist} again}
73 % The new algorithm stores page wide floats together with column floats
74 % in a single |\@deferlist| list. We keep |\@dbldeferlist|
75 % initialised as empty so that packages that are testing for
76 % deferred floats can use the same code for old or new float
77 % handling.

```

```

\gdef\@dbldeferlist{}
\end{macrocode}

```

#### PAGE LAYOUT PARAMETERS

```

78 \newdimen\topmargin
79 \newdimen\oddsidemargin
80 \newdimen\evensidemargin
81 \let\@themargin=\oddsidemargin
82 \newdimen\headheight
83 \newdimen\headsep
84 \newdimen\footskip
85 \newdimen\textheight
86 \newdimen\textwidth
87 \newdimen\columnwidth
88 \newdimen\columnsep
89 \newdimen\columnseprule
90 \newdimen\marginparwidth
91 \newdimen\marginparsep
92 \newdimen\marginparpush

```

`\AtBeginDvi` We use a box register in which to put stuff that must appear before anything else in the .dvi file.

`\@beginvibox` The stuff in the box should not add any typeset material to the page when it is unboxed.

```

93 \newbox\@beginvibox
94 \def \AtBeginDvi #1{%
95 \global \setbox \@beginvibox
96 \vbox{\unvbox \@beginvibox #1}%
97 }

```

```

\@maxdepth This is not the right place to set this; it needs to be set in a class/style file when
\maxdepth is set.
 Also, many settings to \maxdepth should be to \@maxdepth, probably?
98 \newdimen\@maxdepth
99 \@maxdepth = \maxdepth

\paperheight New \paper... registers.
\paperwidth 100 \newdimen\paperheight
101 \newdimen\paperwidth

\if@insert Local switches first:
\if@fcolmade 102 \newif \if@insert
\if@specialpage These should definitely be global:
\if@firstcolumn 103 \newif \if@fcolmade
\if@twocolumn 104 \newif \if@specialpage \@specialpagefalse
\if@twoside
\if@reversemarginpar These should be global but are not always set globally in other files.
\if@mparswitch 105 \newif \if@firstcolumn \@firstcolumntrue
\col@number 106 \newif \if@twocolumn \@twocolumnfalse

Not sure about these: two questions. Should things which must apply to a whole
document be local or global (they probably should be 'preamble only' commands)?
Are these three such things?
107 \newif \if@twoside \@twosidefalse
108 \newif \if@reversemargin \@reversemarginfalse
109 \newif \if@mparswitch \@mparswitchfalse

This counter has been imported from 'multicol'.
110 \newcount \col@number
111 \col@number \@ne

INTERNAL REGISTERS

112 \newcount\@topnum
113 \newdimen\@toproom
114 \newcount\@dbltopnum
115 \newdimen\@dbltoproom
116 \newcount\@botnum
117 \newdimen\@botroom
118 \newcount\@colnum
119 \newdimen\@textmin
120 \newdimen\@fpmin
121 \newdimen\@colht
122 \newdimen\@colroom
123 \newdimen\@pageht
124 \newdimen\@pagedp
125 \newdimen\@mparbottom \@mparbottom\z@
126 \newcount\@currtype
127 \newbox\@outputbox
128 \newbox\@leftcolumn
129 \newbox\@holdpg

130 \def\@thehead{\@oddhead} % initialization
131 \def\@thefoot{\@oddfont}

```

`\clearpage` The tests at the beginning are an experimental attempt to avoid a completely empty page after a `\twocolumn[...]`. This prevents the text from the argument vanishing into a float box, never to be seen again. We hope that it does not produce wrong formatting in other cases.

```

132 \def\clearpage{%
133 \ifvmode
134 \ifnum \@dbltopnum =\m@ne
135 \ifdim \pagetotal <\topskip
136 \hbox{}%
137 \fi
138 \fi
139 \fi
140 \newpage
141 \write\m@ne{}%
142 \vbox{}%
143 \penalty -\@Mi
144 }

```

`\cleardoublepage`

```

145 \def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
146 \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
147 \endkernel

```

`\onecolumn`

```

148 (*2ekernel | ftrace)
149 \def\onecolumn{%
150 \clearpage
151 \global\columnwidth\textwidth
152 \global\hsize\columnwidth
153 \global\linewidth\columnwidth
154 \global\@twocolumnfalse
155 \col@number \@ne
156 \@floatplacement}

```

`\newpage` The two checks at the beginning ensure that an item label or run-in section title immediately before a `\newpage` get printed on the correct page, the one before the page break.

All three tests are largely to make error processing more robust; that is why they all reset the flags explicitly, even when it would appear that this would be done by a `\leavevmode`.

```

157 \def \newpage {%
158 \if@noskipsec
159 \ifx \@nodocument\relax
160 \leavevmode
161 \global \@noskipsecfalse
162 \fi
163 \fi
164 \if@inlabel
165 \leavevmode
166 \global \@inlabelfalse
167 \fi
168 \if@nobreak \@nobreakfalse \everypar{}\fi
169 \par

```

```

170 \vfil
171 \penalty -\@M}

\@emptycol It may be better to use an invisible rule rather than an empty box here.
172 \def \@emptycol {\vbox{}}\penalty -\@M}

\twocolumn There are several bug fixes to the two-column stuff here.
\@topnewpage 173 \def \twocolumn {%
174 \clearpage
175 \global\columnwidth\textwidth
176 \global\advance\columnwidth-\columnsep
177 \global\divide\columnwidth\tw@
178 \global\hsize\columnwidth
179 \global\linewidth\columnwidth
180 \global\@twocolumntrue
181 \global\@firstcolumntrue
182 \col@number \tw@

There is no reason to put a \@dblfloatplacement here since \@topnewpage ig-
nores these settings. The \@floatplacement is needed in case this comes after
some changes.
183 \@ifnextchar [\@topnewpage\@floatplacement
184 }

Note that here, getting a box from the freelist can assume success since this
comes just after a \clearpage.
185 \long\def \@topnewpage [#1]{%
186 \@nodocument
187 \@next\@currbox\@freelist{ }{ }%
188 \global \setbox\@currbox
189 \color@vbox
190 \normalcolor
191 \vbox {%
192 \hsize\textwidth
193 \@parboxrestore
194 \col@number \@ne
195 #1%
196 \vskip -\dbltextfloatsep
197 }%
198 \color@endbox

Added size test and warning message; perhaps we should use an error message.
199 \ifdim \ht\@currbox>\textheight
200 \ht\@currbox \textheight
201 \fi

This next line is not essential but it is more robust to make this value non-zero,
in case of weird errors.

This next bit is what is needed from \@addtodblcol, plus some extra checks
for error trapping.
202 \global \count\@currbox \tw@
203 \@tempdima -\ht\@currbox
204 \advance \@tempdima -\dbltextfloatsep
205 \global \advance \colht \@tempdima
206 \ifx \@dbltoplist \@empty

```



```

207 \else
208 \latexerr{Float(s) lost}\@ehb
209 \let \@dbltoplist \@empty
210 \fi
211 \@cons \@dbltoplist \@currbox

```

This setting of \@dbltopnum is used only to change the typesetting in \@combinedblfloats.

```

212 \global \@dbltopnum \m@ne
213 {*trace}
214 \fl@trace{dbltopnum set to -1 (= \the \@dbltopnum) (topnewpage)}%
215 {/trace}

```

At points such as this we need to check that there is still a minimal amount of room left on the page; this uses an arbitrary small value at present; but note that this value is larger than that used when checking that page is too full of normal floats.

If there is little room left we just force a page-break, OK? This involves producing two empty columns. The second empty column may be produced by \output, in which case an extra, misleading, warning will be generated, OK? (This happens only when there is too little room left on the page for any float.) Otherwise (i.e. if the size is such that it is allowed as a normal float) the extra \@emptycol will be invoked in the second column by the conditional code guarded by the \if@firstcolumn test.

I now think that the cut-off point here should be 3\baselineskip, but we make it a bit less so that 3 lines of text will be allowed, OK?

Since this happens only when there is nothing on the page but the ‘top-box’, the empty box should not cause any problem other than some overfull box messages, which is not entirely misleading.

Here we need two page-ends since both columns need to be empty.

```

216 \ifdim \@colht<2.5\baselineskip
217 \latex@warning@no@line {Optional argument of \noexpand\twocolumn
218 too tall on page \thepage}%
219 \@emptycol
220 \if@firstcolumn
221 \else
222 \@emptycol
223 \fi
224 \else
225 \global \vsize \@colht
226 \global \@colroom \@colht
227 \@floatplacement
228 \fi
229 }

```

\output This needs some small adjustments. We cannot guarantee that the float mechanism will interact correctly with this stuff, but that mechanism does not always work properly with footnotes already.

RmS 91/09/29:

added reset of \par to the output routine. This avoids problems when the output routine is called within a list where \par may be a no-op.

```

230 \output {%
231 \let \par \@@par

```

```

232 \ifnum \outputpenalty<-\@M
233 \@specialoutput
234 \else
235 \@makecol
236 \@opcol

```

Moved to \@opcol: \@floatplacement.

```

237 \@startcolumn

```

This loop could be replaced by an \expandafter tail recursion in \@startcolumn.

```

238 \@whilesw \if@fcolmade \fi
239 {%
240 (*trace)
241 \fl@trace{PAGE: float \if@twocolumn column \else page \fi
242 completed}%
243 (/trace)
244 \@opcol\@startcolumn}%
245 \fi
246 \ifnum \outputpenalty>-\@Miv

```

At points such as this we need to check that there is still a minimal amount of room left on the page; this uses an arbitrary small value at present. If there is little room left we just force a page-break, OK?

This bit is essential only if a float has just been processed so maybe it should be moved; but this is the natural place at which to set the vsize and a test would need to be done anyway. A check has been added to ensure that there really has been a change in the value of \@colroom.

Since this happens only when there is nothing on the page but floats, the empty box should not cause any problem other than some overfull box messages, which is not entirely misleading.

The twocolumn case does not need any extra code here since this is the \output itself; in the second column there will still not be enough room left so \@emptycol will be executed again when the OR is called by the page builder when it gets to the penalty inserted by the first execution. (The page-builder is never invoked whilst the OR is being executed since it builds a inner vlist; thus any conditional code for the two-column case within \output may not get executed with the correct value of \if@firstcolumn.

```

247 \ifdim \@colroom<1.5\baselineskip
248 \ifdim \@colroom<\textheight
249 \@latex@warning@no@line {Text page \thepage\space
250 contains only floats}%
251 \@emptycol
252 % \if@twocolumn
253 % \if@firstcolumn
254 % \else
255 % \@emptycol
256 % \fi
257 % \fi
258 \else
259 \global \vsize \@colroom
260 \fi
261 \else
262 \global \vsize \@colroom
263 \fi

```

```

264 \else
265 \global \vsize \maxdimen
266 \fi
267 }

```

CHANGES TO \@specialoutput:

\* \penalty\z@ changed to \penalty\interlinepenalty so \samepage works properly with figure and table environments.  
(Changed 23 Oct 86)

\* Definition of \@specialoutput changed 26 Feb 88 so \@pageht and \@pagedp aren't changed for a marginal note.  
(Change suggested by Chris Rowley.)

```

268 \gdef\@specialoutput{%
269 \ifnum \outputpenalty>-\@Mii
270 \doclearpage
271 \else
272 \ifnum \outputpenalty<-\@Miii
273 \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
274 \global \setbox\@holdpg \vbox {\unvbox\@cc1v}%
275 \else

```

Note that \boxmaxdepth should not be set here since we wish to record the natural depth of the holdpg box.

This is changed so as to not lose anything, such as writes and marks, which may get into box 255 and should be returned to the list. This should only happen when the first penalty in the mechanism is discarded and therefore \@holdpg should always be void in this case. This can happen because a penalty is discarded whenever there is no box on the list.

It was just: \setbox\@tempboxa \box \@cc1v.

The last box which is removed is the box put there by the double-penalty mechanism. The \unskip then removes the \topskip which is put there since the box is the first on the page.

```

276 \global \setbox\@holdpg \vbox{%
277 \unvbox\@holdpg
278 \unvbox\@cc1v

```

We must now remove the box added by the float mechanism and the \topskip glue therefore added above it by T<sub>E</sub>X.

```

279 \setbox\@tempboxa \lastbox
280 \unskip
281 }%

```

These two are needed as separate dimensions only by \@addmarginpar; for other purposes we put the whole size into \@pageht (see below).

```

282 \@pagedp \dp\@holdpg
283 \@pageht \ht\@holdpg
284 \unvbox \@holdpg
285 \@next\@currbox\@currlist{%
286 \ifnum \count\@currbox>\z@

```

Putting the whole size into \@pageht (see above).

```

287 \advance \@pageht \@pagedp
288 \ifvoid\footins \else

```

```

289 \advance \@pageht \ht\footins
290 \advance \@pageht \skip\footins
291 \advance \@pageht \dp\footins
292 \fi
293 \ifvbox \@kludgeins

```

We want to make the adjustment due to this insert only if the non-star form is used. The \*-form will probably not work with floats, but maybe it still could make some adjustment here even so?

```

294 \ifdim \wd\@kludgeins=\z@
295 \advance \@pageht \ht\@kludgeins
296 (*trace)
297 \fl@trace {Extra size added: \the \ht\@kludgeins}%
298 (/trace)
299 \fi
300 \fi

```

This version puts the inserts back just before the additional material; it could be moved earlier, before unboxing the page-so-far. Neither is guaranteed not to put things on the wrong page. This version is similar to the original version.

```

301 \@reinserts
302 \@addtocurcol
303 \else
304 \@reinserts
305 \@addmarginpar
306 \fi
307 }\@latexbug

```

A 2e change: use `\addpenalty` instead of `\penalty` here. Some penalty is needed to create a potential break-point immediately after the reinserts (or the marginal). Otherwise there can be no possibility to break here and this can cause the reinserts or the marginal to appear on the next page (which is often incorrect). However, if the nobreak flag is true, a `\nobreak` must be correct.

```

308 \ifnum \outputpenalty<\z@
309 \if@nobreak
310 \nobreak
311 \else
312 \addpenalty \interlinepenalty
313 \fi
314 \fi
315 \fi
316 \fi
317 }
318 /2ekernel | fltrace)

```

`\@testwrongwidth` Test if the float box has the wrong width when trying to place it into some area.  
`\f@depth` (Actually the test is for a conventional depth setting rather than for the width of the float. For that reason the box depth was explicitly tailored when the float was created).

```

319 (latexrelease)\IncludeInRelease{2015/01/01}%
320 (latexrelease) {\@testwrongwidth}{float order in 2-column}%
321 (*2ekernel | latexrelease | fltrace)
322 \def\@testwrongwidth #1{%
323 \ifdim\dp#1=\f@depth

```

```

324 (*trace)
325 \fl@trace{\string#1
326 \ifdim\f@depth=\z@ single \else double \fi
327 column float -- ok}%
328 (/trace)
329 \else
330 \global\@testtrue
331 (*trace)
332 \fl@trace{\string#1
333 \ifdim\f@depth=\z@ double \else single \fi
334 column float -- wrong}%
335 (/trace)
336 \fi}%

```

Normally looking for single column floats, which have zero depth.

```

337 \let\f@depth\z@
338 (/2ekernel | latexrelease | fltrace)
339 (latexrelease)\EndIncludeInRelease
340 (latexrelease)\IncludeInRelease{0000/00/00}%
341 (latexrelease) {\@testwrongwidth}{float order in 2-column}%
342 (latexrelease)\let\@testwrongwidth\@undefined
343 (latexrelease)\let\f@depth\@undefined
344 (latexrelease)\EndIncludeInRelease

```

`\@docclearpage` 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.

```

345 (latexrelease)\IncludeInRelease{2015/01/01}{\@docclearpage}%
346 (latexrelease) {float order in 2-column}%
347 (*2ekernel | latexrelease)
348 \def \@docclearpage {%
349 \ifvoid\footins
350 \ifvbox\@kludgeins
351 {\setbox \@tempboxa \box \@kludgeins}%
352 (*trace)
353 \fl@trace {kludgeins box made void}%
354 (/trace)
355 \fi
356 \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
357 \setbox\@tempboxa\box\@cclv
358 \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
359 \global \let \@toplist \@empty
360 \global \let \@botlist \@empty
361 \global \@colroom \@colht
362 \ifx \@currlist\@empty
363 \else

```

```

364 \@latexerr{Float(s) lost}\@ehb
365 \global \let \@currlist \@empty
366 \fi
367 \@makefcolumn\@deferlist
368 \@whiles\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
369 \if@twocolumn
370 \if@firstcolumn
371 \xdef\@deferlist{\@dbltoplist\@deferlist}%
372 \global \let \@dbltoplist \@empty
373 \global \@colht \textheight
374 \begingroup
375 \@dblfloatplacement
376 \@makefcolumn\@deferlist
377 \@whiles\if@fcolmade \fi{\@outputpage
378 \@makefcolumn\@deferlist}%
379 \endgroup
380 \else
381 \vbox{}\clearpage
382 \fi
383 \fi

```

the next line is needed to avoid losing floats in certain circumstances a single call to the original `\doclearpage` will now no longer output all floats.

```

384 \ifx\@deferlist\@empty \else\clearpage \fi
385 \else
386 \setbox\@cclv\vbox{\box\@cclv\vfil}%
387 \@makecol\@opcol
388 \clearpage
389 \fi
390 }%
391 </2ekernel | latexrelease>
392 <latexrelease>\EndIncludeInRelease
393 <latexrelease>\IncludeInRelease{0000/00/00}{\@docclearpage}%
394 <latexrelease> {float order in 2-column}%
395 <latexrelease>\def \@docclearpage {%
396 <latexrelease> \ifvoid\footins

```

We empty any left over kludge insert box here; this is a temporary fix. It should perhaps be applied to one page of cleared floats, but who cares? The whole of this stuff needs completely redoing for many such reasons.

```

397 <latexrelease> \ifvbox\@kludgeins
398 <latexrelease> {\setbox \@tempboxa \box \@kludgeins}%
399 <*trace>
400 <latexrelease> \fl@trace {kludgeins box made void}%
401 </trace>
402 <latexrelease> \fi
403 <latexrelease> \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
404 <latexrelease> \setbox\@tempboxa\box\@cclv
405 <latexrelease> \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
406 <latexrelease> \global \let \@toplist \@empty
407 <latexrelease> \global \let \@botlist \@empty
408 <latexrelease> \global \@colroom \@colht

```

```

409 <latexrelease> \ifx \@currlist\@empty
410 <latexrelease> \else
411 <latexrelease> \@latexerr{Float(s) lost}\@ehb

412 <latexrelease> \global \let \@currlist \@empty
413 <latexrelease> \fi
414 <latexrelease> \@makefcolumn\@deferlist
415 <latexrelease> \@whilesw\if@fcolmade \fi
416 <latexrelease> {\@opcol\@makefcolumn\@deferlist}%
417 <latexrelease> \if@twocolumn
418 <latexrelease> \if@firstcolumn
419 <latexrelease> \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%

420 <latexrelease> \global \let \@dbltoplist \@empty
421 <latexrelease> \global \@colht \textheight
422 <latexrelease> \begingroup
423 <latexrelease> \dblfloatplacement
424 <latexrelease> \@makefcolumn\@dbldeferlist
425 <latexrelease> \@whilesw\if@fcolmade \fi
426 <latexrelease> {\@outputpage\@makefcolumn\@dbldeferlist}%
427 <latexrelease> \endgroup
428 <latexrelease> \else
429 <latexrelease> \vbox{}\clearpage
430 <latexrelease> \fi
431 <latexrelease> \fi
432 <latexrelease> \else
433 <latexrelease> \setbox\@cclv\vbox{\box\@cclv\vfil}%
434 <latexrelease> \@makecol\@opcol
435 <latexrelease> \clearpage
436 <latexrelease> \fi
437 <latexrelease> }%
438 <latexrelease>\EndIncludeInRelease

```

\@opcol Several changes in detail here.

```

439 (*2kernel | fltrace)
440 \def \@opcol {%
441 \if@twocolumn
442 \outputdblcol
443 \else
444 \outputpage
445 (*trace)
446 \fl@trace{PAGE: one column (float? see above) page completed}%
447 </trace>

```

Not needed since it comes after \@outputpage:

```

448 % \global\@colht\textheight
449 \fi

```

These do not need to be done every time \@opcol is used: they should be grouped together since they all need to be done at the end of the non-special output routine, or at the end of a clearpage one.

```

450 \global \@mparbottom \z@ \global \@textfloatsheight \z@
451 \floatplacement
452 }
453 </2kernel | fltrace>

```

`\@makecol` We must rewrite this macro to allow for variations in page-makeup required by changes in page-length.

This uses a different macro if a special-length column is being produced.

```
454 (*2kernel)
455 \gdef \@makecol {%
456 \ifvoid\footins
457 \setbox\@outputbox \box\@cclv
458 \else
459 \setbox\@outputbox \vbox {%
```

This `\boxmaxdepth` setting is to ensure that deep footnotes do not overwrite the footer (on account of the negative skip added later): it should use `\@maxdepth` otherwise the change is pointless when there are footnotes.

But see also its use when combining floats.

```
460 \boxmaxdepth \@maxdepth

461 % \@tempdima\dp\@cclv
462 \unvbox \@cclv
463 % \vskip-\@tempdima
464 \vskip \skip\footins

465 \color@begingroup
466 \normalcolor
467 \footnoterule
468 \unvbox \footins
469 \color@endgroup
470 }%
471 \fi
```

The h floats have now been finally committed to this page so we can reset their list. The top and bottom floats are then added to the page.

```
472 \let\@elt\relax
473 \xdef\@freelist{\@freelist\@midlist}%

474 \global \let \@midlist \@empty
475 \@combinefloats
```

The variations start here in case `\enlargethispage` has been used.

```
476 \ifvbox\@kludgeins
477 \@makespecialcolbox
478 \else
```

This extra reboxing is only needed to add the `\@texttop` and `\@textbottom` but this could be done earlier, when the floats are added.

The `\boxmaxdepth` resetting here will have no effect unless `\@textbottom` ends with a box or rule. So is this (or possibly `\@maxdepth`) the correct value?

The `\vskip -\dimen@` ensures that the visible depth of the box does not affect the placement of anything on the page. Thus very deep pages will overprint the footer; but these should have been prevented by suitable settings of the maxdepths at appropriate times.

If `\@textbottom` ends with a box or rule of non-zero depth then this skip adjustment should be done again after it.

I think that the final boxing of the main text page could have a common ending which may make it simpler to see what is going on.



This needs further investigation, especially in the ‘special case’.

Also, the `\boxmaxdepth` setting here affects what happens within `\@texttop` and `\@textbottom`, should it? Is it needed at all?

RmS 91/10/22: Replaced `\dimen128` by `\dimen@`.

```
479 \setbox\@outputbox \vbox to\@colht {%
480 % \boxmaxdepth \maxdepth %??
481 \@texttop
482 \dimen@ \dp\@outputbox
483 \unvbox \@outputbox
484 \vskip -\dimen@
485 \@textbottom
486 }%
487 \fi
488 \global \maxdepth \@maxdepth
489 }
```

`\@reinserts` This is the code which reinserts the inserts. It puts them all in one place; this can make some of them come out on the wrong page. It has been put into a separate macro to expedite experimentation.

```
490 \gdef \@reinserts{%
491 \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
492 \ifvbox\@kludgeins\insert\@kludgeins
493 {\unvbox\@kludgeins}\fi
494 }
495 </2kernel>
```

`\@makespecialcolbox` This implements certain variations in page-makeup.

```
496 <*2kernel | fltrace>
497 \gdef \@makespecialcolbox {%
498 <*trace>
499 \fl@trace{Kludgeins ht \the\ht\@kludgeins\space
500 dp \the\dp\@kludgeins\space
501 wd \the\wd\@kludgeins}%
502 </trace>
```

First we find the natural height of the column.

See above for discussion of what is happening here.

This needs further investigation, especially in this ‘special case’.

```
503 \setbox\@outputbox \vbox {%
504 \@texttop
505 \dimen@ \dp\@outputbox
506 \unvbox\@outputbox
507 \vskip-\dimen@
508 }%
509 \@tempdima \@colht
510 \ifdim \wd\@kludgeins>\z@
```

Note that in this case (the \*-version), the height of the `\@kludgeins` box is not used since its value is somewhat arbitrary: it need only be big enough to ensure that the page-break is not taken prematurely.

Here we calculate how much vertical space needs to be added in order to enable the column to fit into a box of size `\@colht` using the best information we have about the amount of shrink available (another thing which is known internally about a box, but cannot be accessed at the  $\TeX$  level!).

This needs T<sub>E</sub>X3 otherwise `\pageshrink` is zero anyway; it may not be exactly the figure we wish as it is the total available from the all the material collected before the page-break decision is made. It will, we think, always be an overestimate of the actual shrink in the box; therefore this should always force the shortest possible column with the possibility of an overfull box.

This should work for 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.

```

511 \advance \@tempdima -\ht\@outputbox
512 \advance \@tempdima \pageshrink
513 (*trace)
514 \fl@trace {Natural ht of col: \the \ht\@outputbox}%
515 \fl@trace {\string \@colht: \the \@colht}%
516 \fl@trace {Pageshrink added: \the \pageshrink}%
517 \fl@trace {Hence, space added: \the \@tempdima}%
518 </trace>
519 \setbox\@outputbox \vbox to \@colht {%
520 % \boxmaxdepth \maxdepth
521 \unvbox\@outputbox
522 \vskip \@tempdima
523 \@textbottom
524 }%
```

For the unstarred version, the final size of the page is precisely specified. Therefore, at least for the flush-bottom case, we need to ensure that, visually, it has this size exactly.

Thus we calculate this size and set the material in a box of this size, which is then put into a box of size `\@colht` with `\vss` at the bottom.

```

525 \else
526 \advance \@tempdima -\ht\@kludgeins
527 (*trace)
528 \fl@trace {Natural ht of col: \the \ht\@outputbox}%
529 \fl@trace {\string \@colht: \the \@colht}%
530 \fl@trace {Extra size added: -\the \ht \@kludgeins}%
531 \fl@trace {Hence, height of inner box: \the \@tempdima}%
532 \fl@trace {Max? pageshrink available: \the \pageshrink}%
533 </trace>
```

This type of final packaging could be done always; this may simplify all of this page-makeup.

It is not necessary to set `\boxmaxdepth` here since the `\@outputbox` ends with glue.

```

534 \setbox \@outputbox \vbox to \@colht {%
535 \vbox to \@tempdima {%
536 \unvbox\@outputbox
537 \@textbottom}%
538 \vss}%
539 \fi
```

Finally we need to explicitly make the insert box void.

```

540 {\setbox \@tempboxa \box \@kludgeins}%
541 (*trace)
542 \fl@trace {kludgeins box made void}%
543 </trace)
544 }
545 </2ekernel | fltrace)

\@texttop These do nothing as a default.
\@textbottom 546 (*2ekernel)
547 \let \@texttop \relax
548 \let \@textbottom \relax

\@resetactivechars RmS 93/09/06: added hook to protect against certain active characters in the
\@activechar@info output routine. Default checks are for active space and end-of-line.
549 \def\@activechar@info #1{%
550 \@latex@info@no@line {Active #1 character found while
551 output routine is active
552 \MessageBreak
553 This may be a bug in a package file
554 you are using}%
555 }

Do not put any spaces in this next bit!

556 \begingroup
557 \obeylines\obeyspaces%
558 \catcode'\'\active%
559 \gdef\@resetactivechars{%
560 \def^^M{\@activechar@info{EOL}\space}%
561 \def {\@activechar@info{space}\space}%
562 \let'\active@math@prime}%
563 \endgroup

\@outputpage The \color@hbox hooks here are used to avoid putting just a colour special into
\@shipoutsetup an otherwise empty box (in a header or footer). These boxes are often set to be
\@writesetup completely empty and so adding a special produces a very underfull box message.
There has been extensive tidying up of the old code here; including the removal
of a level of grouping.
The setting of \protect immediately before the \shipout is needed so that
protected commands within \writes are handled correctly.
Within shipout's vbox it is reset to its default value, \relax.
Resetting it to its default value after the shipout has been completed (and the
contents of the writes have been expanded) must be done by use of \aftergroup.
This is because it must have the value \relax before macros coming from other
uses of \aftergroup within this box are expanded.
Putting this into the \aftergroup token list does not affect the definition
used in expanding the \writes because the aftergroup token list is only con-
structed when popping the save-stack, it is not expanded until after the shipout
is completed.
Question: should things from an \aftergroup within the shipped out box be
executed in the environment set up for the writes, or after it finishes?
A lot of this code has been in-lined tp prevent mis-use of internal commands
as hooks.

```

```

564 \def\@outputpage{%
565 \begingroup % the \endgroup is put in by \aftergroup

```

Now all the set-up stuff has been in-lined for Frank.

First the stuff for the writes.

From here ... was in the command \@writesetup.

```

566 \let \protect \noexpand

```

RmS 93/08/19: Redefined accents to allow changes in font encoding; but exactly why was this needed?

The \catcode'\ = 10 was removed as it was considered useless (presumably because nothing gets tokenised during shipout).

This was put in as some error produced active spaces in a mark, I think.

Why was the hyphen reset?

```

567 \@resetactivechars

```

If a page break happens between the start of a list and its first item the @newlist will be true and this will mess up any list that is used in the header or footer of the page. So we have to reset that flag.

```

568 \global\let\@if@newlist\if@newlist

```

```

569 \global\@newlistfalse

```

This next hook replaces the following:

```

\let-\@dischyph
\let'\@acci\let'\@accii\let\=\@acciii
\let\\@normalcr
\let\par\@par %% 15 Sep 87 (this was once inside the box)

```

and it does more than they did; in particular it sets:

```

\parindent\z@
\parskip\z@skip
\everypar{}%
\leftskip\z@skip
\rightskip\z@skip
\parfillskip\@flushglue
\lineskip\normallineskip
\baselineskip\normalbaselineskip
\sloppy

```

```

570 \@parboxrestore

```

... to here was in the command \@writesetup.

```

571 \shipout \vbox{%

```

```

572 \set@typeset@protect

```

```

573 \aftergroup \endgroup

```

```

574 \aftergroup \set@typeset@protect

```

```

575 % correct? or just restore by ending

```

```

576 % the group?

```

This first bit has been moved inside the shipped out box.

Now the setup inside the shipped out box; this should contain all the stuff that could only affect typesetting; other stuff may need to be reset for the writes also.

From here ... was in the command \@shipoutsetup.

```

577 \if@specialpage
578 \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
579 \fi
580 \if@twoside
581 \ifodd\count\z@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
582 \let\@themargin\oddsidemargin
583 \else \let\@thehead\@evenhead
584 \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
585 \fi
586 \fi

```

The rest was always inside the box.

RmS 91/08/15: aded this line:

```
587 \reset@font
```

RmS 93/08/06 Added `\lineskiplimit=0pt` to guard against it being nonzero: e.g. by `\offinterlineskip` being in effect.

There are probably lots of other things that may need resetting.

```
588 \normalsize
```

Reset the space factors.

```
589 \normalsfcodes
```

Reset these here (previously reset separately for head and foot)

```

590 \let\label\@gobble
591 \let\index\@gobble
592 \let\glossary\@gobble
593 \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@

```

... to here was in the command `\@shipoutsetup`.

```

594 \@beginndvi
595 \vskip \topmargin
596 \moveright\@themargin \vbox {%
597 \setbox\@tempboxa \vbox to\headheight{%
598 \vfil
599 \color@hbox
600 \normalcolor
601 \hb@xt@\textwidth{\@thehead}%
602 \color@endbox
603 }%
604 \dp\@tempboxa \z@
605 \box\@tempboxa
606 \vskip \headsep
607 \box\@outputbox
608 \baselineskip \footskip
609 \color@hbox
610 \normalcolor
611 \hb@xt@\textwidth{\@thefoot}%
612 \color@endbox
613 }%
614 }%

```

`\endgroup` now inserted by `\aftergroup`

Restore `\if@newlist`

```
615 \global\let\if@newlist\@if@newlist
```

```

616 \global \colht \textheight
617 \stepcounter{page}%

```

It is now clear that this does something useful, thanks to Piet van Oostrum. It is needed because a float page is made without using TeX's page-builder; thus the output routine is never called so the marks are not updated.

```

618 \let\firstmark\botmark
619 }

```

`\@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.

```

620 \def \@begindivi{%
621 \unvbox \@begindivibox
622 \global\let \@begindivi \@empty
623 }

```

`\@combinefloats` The `\boxmaxdepth` setting here was not made local to a box so was dangerous. It is needed only within the box made by `\@cflt` (and not normally even there), so `\@cflb` it has been moved there; this also agrees with the original pseudocode.

```

624 \def \@combinefloats {%
625 % \boxmaxdepth \maxdepth
626 \ifx \@toplist\@empty \else \@cflt \fi
627 \ifx \@botlist\@empty \else \@cflb \fi
628 }

629 \def \@cflt{%
630 \let \@elt \@comflelt
631 \setbox\@tempboxa \vbox{}%
632 \@toplist
633 \setbox\@outputbox \vbox{%
634 \boxmaxdepth \maxdepth
635 \unvbox\@tempboxa
636 \vskip -\floatsep
637 \topfigrule
638 \vskip \textfloatsep
639 \unvbox\@outputbox
640 }%
641 \let\@elt\relax
642 \xdef\@freelist{\@freelist\@toplist}%
643 \global\let\@toplist\@empty
644 }

645 \def \@cflb {%
646 \let\@elt\@comflelt
647 \setbox\@tempboxa \vbox{}%
648 \@botlist
649 \setbox\@outputbox \vbox{%
650 \unvbox\@outputbox
651 \vskip \textfloatsep
652 \botfigrule
653 \unvbox\@tempboxa
654 \vskip -\floatsep
655 }%

```

```

656 \let\@elt\relax
657 \xdef\@freelist{\@freelist\@botlist}%
658 \global \let \@botlist\@empty
659 }

```

```

\@comflelt
\@comdblfilelt
\@combinedblfloats
660 \def\@comflelt#1{\setbox\@tempboxa
661 \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
662 \def\@comdblfilelt#1{\setbox\@tempboxa
663 \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
664 \def \@combinedblfloats{%
665 \ifx \@dbltoplist \@empty
666 \else
667 \setbox\@tempboxa \vbox{}%
668 \let \@elt \@comdblfilelt
669 \@dbltoplist
670 \let \@elt \relax
671 \xdef \@freelist {\@freelist\@dbltoplist}%
672 \global\let \@dbltoplist \@empty
673 \setbox\@outputbox \vbox to\textheight

```

The setting of `\boxmaxdepth` here has no effect since the `\@outputbox` should already have depth zero. Even so, it would have no effect on the layout of the page.

```

674 {\boxmaxdepth\maxdepth %% probably not needed, CAR
675 \unvbox\@tempboxa\vskip-\dblfloatsep

```

Here we need different typesetting if the top float comes from `\@topnewpage`.

```

676 \ifnum \@dbltopnum>\m@ne
677 \dblfigrule
678 \fi
679 \vskip \dbltextfloatsep
680 \box\@outputbox
681 }%
682 \fi
683 }
684 </2kernel>

```

`\@startcolumn` We could combine (most of) these two into `\@startcol <list>`. Note that `\@startdblcolumn` `\@xstartcol` was only used once (i.e. in `\@startcolumn`); it has therefore been removed. This is not quite as efficient but it now has the same structure as `\@startdblcolumn`.

The empty-list test has been moved to `\@tryfcolumn`.

```

685 < *2kernel | fltrace >
686 \def \@startcolumn {%
687 \global \@colroom \@colht
688 \@tryfcolumn \@deferlist
689 \if@fcolmade
690 < *trace >
691 \fl@trace{PAGE: float \if@twocolumn column \else page \fi
692 completed}%
693 </trace >
694 \else

```

```

695 \begingroup
696 \let \reserved@b \@deferlist
697 \global \let \@deferlist \@empty
698 \let \@elt \@scolelt
699 \reserved@b
700 \endgroup
701 \fi
702 }

```

This one does not need to set \@colht.

```

703 </2ekernel | fltrace)
704 (latexrelease | fltrace)\IncludeInRelease{2015/01/01}%
705 (latexrelease | fltrace) {\@startdblcolumn}{float order in 2-column}%
706 (*2ekernel | latexrelease | fltrace)
707 \def \@startdblcolumn {%
708 \@tryfcolumn \@deferlist
709 \if@fcolmade
710 (fltrace) \fl@trace{PAGE: double float page completed}%
711 \else
712 \begingroup
713 \let \reserved@b \@deferlist
714 \global \let \@deferlist \@empty
715 \let \@elt \@sdblcolelt
716 \reserved@b
717 \endgroup
718 \fi
719 }%
720 </2ekernel | latexrelease | fltrace)
721 (latexrelease | fltrace)\EndIncludeInRelease
722 (latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
723 (latexrelease | fltrace) {\@startdblcolumn}{float order in 2-column}%
724 (latexrelease | fltrace)\def \@startdblcolumn {%

```

Not needed since this always comes after \@outputpage:

```

725 (latexrelease | fltrace)% \global \@colht \textheight
726 (latexrelease | fltrace) \@tryfcolumn \@dbldeferlist
727 (latexrelease | fltrace) \if@fcolmade
728 (*trace)
729 (latexrelease | fltrace) \fl@trace{PAGE: double float page completed}%
730 </trace)
731 (latexrelease | fltrace) \else
732 (latexrelease | fltrace) \begingroup
733 (latexrelease | fltrace) \let \reserved@b \@dbldeferlist
734 (latexrelease | fltrace) \global \let \@dbldeferlist \@empty
735 (latexrelease | fltrace) \let \@elt \@sdblcolelt
736 (latexrelease | fltrace) \reserved@b
737 (latexrelease | fltrace) \endgroup
738 (latexrelease | fltrace) \fi
739 (latexrelease | fltrace)}%
740 (latexrelease | fltrace)\EndIncludeInRelease
741 (*2ekernel | fltrace)

```

\@tryfcolumn Now tests if its list is empty before any further exertion.



```

742 \def \@tryfcolumn #1{%
743 \global \@fcolmadefalse
744 \ifx #1\@empty
745 \else
746 (*trace)
747 \fl@trace{PAGE: try float \if@twocolumn column/page\else page\fi
748 ---\string #1}%
749 \fl@trace{----- \string #1: #1}%
750 /trace)

751 \xdef\@trylist{#1}%
752 \global \let \@failedlist \@empty
753 \begingroup
754 \let \@elt \@xtryfc \@trylist
755 \endgroup
756 \if@fcolmade
757 \@vtryfc #1%
758 \fi
759 \fi
760 }
761 /2ekernel | fltrace)

762 (*2ekernel)

\@scolelt
763 \def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}

\@sdblcolelt
764 \def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}

\@vtryfc
765 \def\@vtryfc #1{%
766 \global\setbox\@outputbox\vbox{}%
767 \let\@elt\@wtryfc
768 \@flsucceed
769 \global\setbox\@outputbox \vbox to\@colht{%
770 \vskip \@fptop
771 \vskip -\@fpsep
772 \unvbox \@outputbox
773 \vskip \@fpbot}%
774 \let\@elt\relax
775 \xdef #1{\@failedlist\@flfail}%
776 \xdef\@freelist{\@freelist\@flsucceed}}

\@wtryfc
777 \def\@wtryfc #1{%
778 \global\setbox\@outputbox\vbox{%
779 \unvbox\@outputbox
780 \vskip\@fpsep
781 \box #1}}

\@xtryfc
782 /2ekernel)
783 (latexrelease)\IncludeInRelease{2015/01/01}{\@xtryfc}%

```

```

784 <latexrelease> {float order in 2-column}%
785 <*/2ekernel | latexrelease>
786 \def\@xtryfc #1{%
787 \@next\reserved@a\@trylist{ }{ }%
788 \@currtype \count #1%
789 \divide\@currtype\@xxxii
790 \multiply\@currtype\@xxxii
791 \@bitor \@currtype \@failedlist
792 \@testfp #1%
793 \@testwrongwidth #1%
794 \ifdim \ht #1>\@colht
795 \@testtrue
796 \fi
797 \if@test
798 \@cons\@failedlist #1%
799 \else
800 \@ytryfc #1%
801 \fi}%
802 </2ekernel | latexrelease>
803 <latexrelease>\EndIncludeInRelease
804 <latexrelease>\IncludeInRelease{0000/00/00}{\@xtryfc}%
805 <latexrelease> {float order in 2-column}%
806 <latexrelease>\def\@xtryfc #1{%
807 \@next\reserved@a\@trylist{ }{ }%
808 \@currtype \count #1%
809 \divide\@currtype\@xxxii
810 \multiply\@currtype\@xxxii
811 \@bitor \@currtype \@failedlist
812 \@testfp #1%
813 \ifdim \ht #1>\@colht
814 \@testtrue
815 \fi
816 \if@test
817 \@cons\@failedlist #1%
818 \else
819 \@ytryfc #1%
820 \fi}%
821 <latexrelease>\EndIncludeInRelease
822 <*/2ekernel>

```

\@ytryfc

```

823 \def\@ytryfc #1{%
824 \begingroup
825 \gdef\@flsucceed{\@elt #1}%
826 \global\let\@flfail\@empty
827 \@tempdima\ht #1%
828 \let\@elt\@ztryfc
829 \@trylist
830 \ifdim \@tempdima >\@fpmmin
831 \global\@fcolmadetrue
832 \else
833 \@cons\@failedlist #1%
834 \fi

```

```

835 \endgroup
836 \if@fcolmade
837 \let\@elt\@gobble
838 \fi}

```

\@ztryfc

```

839 </2ekernel>
840 <latexrelease>\IncludeInRelease{2015/01/01}{@ztryfc}%
841 <latexrelease> \float order in 2-column}%
842 <*2ekernel | latexrelease>
843 \def\@ztryfc #1{%
844 \@tempcnta\count #1%
845 \divide\@tempcnta\@xxxii
846 \multiply\@tempcnta\@xxxii
847 \@bitor \@tempcnta {\@failedlist \@flfail}%
848 \@testfp #1%
 not in fixfloats?
849 \@testwrongwidth #1%
850 \@tempdimb\@tempdima
851 \advance\@tempdimb\ht #1%
852 \advance\@tempdimb\@fpsep
853 \ifdim \@tempdimb >\@colht
854 \@testtrue
855 \fi
856 \if@test
857 \@cons\@flfail #1%
858 \else
859 \@cons\@flsucceed #1%
860 \@tempdima\@tempdimb
861 \fi}%
862 </2ekernel | latexrelease>
863 <latexrelease>\EndIncludeInRelease
864 <latexrelease>\IncludeInRelease{0000/00/00}{@ztryfc}%
865 <latexrelease> \float order in 2-column}%
866 <latexrelease>\def\@ztryfc #1{%
867 <latexrelease> \@tempcnta \count#1%
868 <latexrelease> \divide\@tempcnta\@xxxii
869 <latexrelease> \multiply\@tempcnta\@xxxii
870 <latexrelease> \@bitor \@tempcnta {\@failedlist \@flfail}%
871 <latexrelease> \@testfp #1%
872 <latexrelease> \@tempdimb\@tempdima
873 <latexrelease> \advance\@tempdimb \ht#1%
874 <latexrelease> \advance\@tempdimb\@fpsep
875 <latexrelease> \ifdim \@tempdimb >\@colht
876 <latexrelease> \@testtrue
877 <latexrelease> \fi
878 <latexrelease> \if@test
879 <latexrelease> \@cons\@flfail #1%
880 <latexrelease> \else
881 <latexrelease> \@cons\@flsucceed #1%
882 <latexrelease> \@tempdima\@tempdimb
883 <latexrelease> \fi}%
884 <latexrelease>\EndIncludeInRelease

```

The major changes for float suppression and the changes to the float mechanism to make it conform to the documentation are in these next macros.

`\@addtobot` Lots of changes.

```

885 (*2kernel | fltrace)
886 \def \@addtobot {%
887 (*trace)
888 \fl@trace{***Start addtobot}%
889 (/trace)
890 \@getfpsbit 4\relax
891 (*trace)
892 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi bot:
893 \the \@fpstype}%
894 (/trace)
895 \ifodd \@tempcnta
896 \@flsetnum \@botnum
897 \ifnum \@botnum>\z@
898 \@tempswafalse
899 \@flcheckspace \@botroom \@botlist
900 \if@tempswa

```

This next line means that this page is produced with box 255 having depth zero, rather than the normal maxdepth: is this needed, useful?

```

901 \global \maxdepth \z@
902 \@flupdates \@botnum \@botroom \@botlist
903 (*trace)
904 \fl@trace{colroom (after-bot) = \the \@colroom}%
905 \fl@trace{colnum (after-bot) = \the \@colnum}%
906 \fl@trace{botnum (after-bot) = \the \@botnum}%
907 \fl@trace{***Success: bot}%
908 (/trace)
909 \inserttrue
910 \fi
911 (*trace)
912 \else
913 \fl@trace{Fail: botnum = \the \@botnum:
914 fpstype \the \@fpstype=ORD?}%
915 \ifnum \@fpstype<\sift@n
916 \fl@trace{ERROR: !b float not successful (addtobot)}%
917 \fi
918 (/trace)
919 \fi
920 \fi
921 }

```

`\@addtotoporbot` Lots of changes.

```

922 \def \@addtotoporbot {%
923 (*trace)
924 \fl@trace{***Start addtotoporbot}%
925 (/trace)
926 \@getfpsbit \tw@
927 (*trace)
928 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi top:
929 \the \@fpstype}%

```

```

930 </trace>
931 \ifodd \@tempcnta
932 \@flsetnum \@topnum
933 \ifnum \@topnum>\z@
934 \@tempwafalse
935 \@flcheckspace \@toproom \@toplist
936 \if@tempswa
937 \@bitor\@currtype{\@midlist\@botlist}%
938 (*trace)
939 \fl@trace{(mid+bot)list: \@midlist, \@botlist:
940 (addtotoporbot-before)}%
941 </trace>
942 \if@test
943 (*trace)
944 \fl@trace{type already on list: mid or bot---sent to addtobot}%
945 </trace>
946 \else
947 \@flupdates \@topnum \@toproom \@toplist
948 (*trace)
949 \fl@trace{colroom (after-top) = \the \@colroom}%
950 \fl@trace{colnum (after-top) = \the \@colnum}%
951 \fl@trace{topnum (after-top) = \the \@topnum}%
952 \fl@trace{***Success: top}%
953 </trace>
954 \@inserttrue
955 \fi
956 \fi
957 (*trace)
958 \else
959 \fl@trace{Fail: topnum = \the \@topnum: fpstype
960 \the \@fpstype=ORD?}%
961 \ifnum \@fpstype<\sixt@n
962 \fl@trace{ERROR: !t float not successful (addtotoporbot)}%
963 \fi
964 </trace>
965 \fi
966 \fi
967 \if@insert
968 \else
969 (*trace)
970 \fl@trace{sent to addtobot (addtotoporbot)}%
971 </trace>
972 \@addtobot
973 \fi
974 }
975 </2ekernel | fltrace>

```

\@addtocurcol Lots of changes.

```

976 <latexrelease | fltrace | flafter>\IncludeInRelease{2015/01/01}%
977 <latexrelease | fltrace | flafter> {\@addtocurcol}{float order in 2-column}%
978 <*2ekernel | latexrelease | fltrace | flafter>
979 \def \@addtocurcol {%
980 (*trace)
981 \fl@trace{***Start addtocurcol}%

```

```

982 </trace>
983 \@insertfalse
984 \@setfloattypecounts
985 \ifnum \@fpstype=8
986 < *trace>
987 \fl@trace{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
988 </trace>
989 \else
990 \ifnum \@fpstype=24
991 < *trace>
992 \fl@trace{fpstype p only (addtocurcol): \the \@fpstype = 24?}%
993 </trace>
994 \else
995 \@flsettextmin

```

This is a new adjustment which is quite a major change in functionality; but it implements the documentation. Note that `\@reqcolroom` will include the whole of the page-so-far, and hence includes `\@textfloatsheight` of floats, so before comparing it with `\@textmin`, we add this to `\@textmin` also.

```

996 < *trace>
997 \fl@trace{textfloatsheight (before) = \the \@textfloatsheight}%
998 </trace>
999 \advance \@textmin \@textfloatsheight
1000 \@reqcolroom \@pageht

```

This line must be removed since `\@specialoutput` changed.

```

1001 % \advance \@reqcolroom \@pagedp
1002 < *trace>
1003 \fl@trace{textmin + textfloatsheight: \the \@textmin}%
1004 \fl@trace{page-so-far: \the \@reqcolroom}%
1005 </trace>
1006 \ifdim \@textmin>\@reqcolroom
1007 \@reqcolroom \@textmin
1008 < *trace>
1009 \fl@trace{ORD? textmin being used}%
1010 </trace>
1011 \fi
1012 \advance \@reqcolroom \ht\@currbox
1013 < *trace>
1014 \fl@trace{float size = \the \ht \@currbox (addtocurcol)}%
1015 \fl@trace{colroom = \the \@colroom (addtocurcol)}%
1016 \fl@trace{reqcolroom = \the \@reqcolroom (addtocurcol)}%
1017 </trace>
1018 \ifdim \@colroom>\@reqcolroom
1019 \@flsetnum \@colnum
1020 \ifnum \@colnum>\z@
1021 \@bitor\@currtype\@deferlist

```

We need to defer the float also if its width doesn't fit.

```

1022 \@testwrongwidth\@currbox
1023 < *trace>
1024 \fl@trace{deferlist: \@deferlist: (addtocurcol-before)}%
1025 </trace>
1026 \if@test

```

```

1027 (*trace)
1028 \fl@trace{type already on list: defer (addtocurcol)}%
1029 (/trace)
1030 \else
1031 \@bitor\@currtype\@botlist
1032 (*trace)
1033 \fl@trace{botlist: \@botlist: (addtocurcol-before)}%
1034 (/trace)
1035 \if@test
1036 (*trace)
1037 \fl@trace{type already on list: bot---sent to addtobot}%
1038 (/trace)
1039 \@addtobot
1040 \else
1041 (*trace)
1042 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi
1043 here: \the \@fpstype}%
1044 (/trace)
1045 \ifodd \count\@currbox
1046 \advance \@reqcolroom \intextsep
1047 \ifdim \@colroom>\@reqcolroom
1048 \global \advance \@colnum \m@ne
1049 \global \advance \@textfloatsheight \ht\@currbox

This may sometimes give an overestimate.
1050 \global \advance \@textfloatsheight 2\intextsep
1051 \@cons \@midlist \@currbox
1052 (*trace)
1053 \fl@trace{***Success: here}%
1054 \fl@trace{textfloatsheight (after-here) =
1055 \the \@textfloatsheight}%
1056 \fl@trace{colnum (after-here) = \the \@colnum}%
1057 (/trace)

CHANGE TO \@addtocurcol:
\penalty\z@ changed to \penalty\interlinepenalty so \samepage works
properly with figure and table environments. (Changed 23 Oct 86)
There is also an \addpenalty\interlinepenalty above.
Since in 2e \samepage is no longer supported, these could be removed.
Although it is best to use \addvspace in case two h floats come together, this
makes other spacing more difficult to adjust; whereas if a user specifies two h floats
together then they can more easily get the spacing correct by ad hoc commands.
It is necessary to adjust for the addition of \parskip here in case the float is
added between paragraphs (i.e. when in vertical mode).
If the nobreak switch is true we need to reset it and clear \everypar since the
float may not reset the flag and cannot reset the \everypar globally.
Typesetting starts here (we are in vertical mode).
1058 \if@nobreak
1059 \nobreak
1060 \@nobreakfalse
1061 \everypar{}%
1062 \else
1063 \addpenalty \interlinepenalty
1064 \fi

```

```

1065 \vskip \intextsep
1066 \box\@currbox
1067 \penalty\interlinepenalty
1068 \vskip\intextsep
1069 \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi

Typesetting ends here.

1070 \outputpenalty \z@
1071 \inserttrue
1072 (*trace)
1073 \else
1074 \fl@trace{Fail---no room at 2nd test of colroom
1075 (addtocurcol \string\intextsep)}%
1076 (/trace)
1077 \fi
1078 \fi
1079 \if@insert
1080 \else

Next set of docstrip guards are a bit weird, essentially \@addtotoporbot ends
up inside the kernel and the fltrace package and \@addtobot shows up in the
flafter package. Guess that could have been done a bit more obvious :-

1081 (*!2kernel | fltrace | latexrelease)
1082 (*trace)
1083 \fl@trace{not here: sent to addtotoporbot}%
1084 (/trace)
1085 \@addtotoporbot
1086 (/!2kernel | fltrace | latexrelease)
1087 (*!2kernel&!fltrace&!latexrelease)
1088 (*trace)
1089 \fl@trace{not here: sent to addtobot}%
1090 (/trace)
1091 \@addtobot
1092 (/!2kernel&!fltrace&!latexrelease)
1093 \fi
1094 \fi
1095 \fi
1096 (*trace)
1097 \else
1098 \fl@trace{Fail: colnum = \the \@colnum:
1099 fpstype \the \@fpstype=ORD?}%
1100 \ifnum \@fpstype<\sixt@n
1101 \fl@trace{ERROR: BANG float not successful (addtocurcol)}%
1102 \fi
1103 (/trace)
1104 \fi
1105 (*trace)
1106 \else
1107 \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1108 (addtocurcol)}%
1109 (/trace)
1110 \fi
1111 \fi
1112 \fi
1113 \if@insert

```



```

1114 \else
1115 \@resethfps
1116 (*trace)
1117 \fl@trace{put on deferlist (addtocurcol)}%
1118 /trace)
1119 \@cons\@deferlist\@currbox
1120 (*trace)
1121 \fl@trace{deferlist: \@deferlist: (addtocurcol-after)}%
1122 /trace)
1123 \fi
1124 }%
1125 /2kernel | latexrelease | fltrace | flafter)
1126 (latexrelease | fltrace | flafter)\EndIncludeInRelease
1127 (latexrelease | fltrace | flafter)\IncludeInRelease{0000/00/00}%
1128 (latexrelease | fltrace | flafter) {\@addtocurcol}{float order in 2-column}%
1129 (latexrelease | fltrace | flafter)\def \@addtocurcol {%
1130 (*trace)
1131 (latexrelease | fltrace | flafter) \fl@trace{***Start addtocurcol}%
1132 /trace)
1133 (latexrelease | fltrace | flafter) \@insertfalse
1134 (latexrelease | fltrace | flafter) \@setfloattyperecounts
1135 (latexrelease | fltrace | flafter) \ifnum \@fpstype=8
1136 (*trace)
1137 (latexrelease | fltrace | flafter) \fl@trace{fpstype !p only (addtocurcol):
1138 (latexrelease | fltrace | flafter) \the \@fpstype = 8?}%
1139 /trace)
1140 (latexrelease | fltrace | flafter) \else
1141 (latexrelease | fltrace | flafter) \ifnum \@fpstype=24
1142 (*trace)
1143 (latexrelease | fltrace | flafter) \fl@trace{fpstype p only (addtocurcol):
1144 (latexrelease | fltrace | flafter) \the \@fpstype = 24?}%
1145 /trace)
1146 (latexrelease | fltrace | flafter) \else
1147 (latexrelease | fltrace | flafter) \@flsettextmin

This is a new adjustment which is quite a major change in functionality; but it
implements the documentation. Note that \@reqcolroom will include the whole
of the page-so-far, and hence includes \@textfloatsheight of floats, so before
comparing it with \@textmin, we add this to \@textmin also.

1148 (*trace)
1149 (latexrelease | fltrace | flafter) \fl@trace{textfloatsheight (before) =
1150 (latexrelease | fltrace | flafter) \the \@textfloatsheight}%
1151 /trace)
1152 (latexrelease | fltrace | flafter) \advance \@textmin \@textfloatsheight
1153 (latexrelease | fltrace | flafter) \@reqcolroom \@pageht

This line must be removed since \@specialoutput changed.

1154 % \advance \@reqcolroom \@pagedp
1155 (*trace)
1156 (latexrelease | fltrace | flafter) \fl@trace{textmin + textfloatsheight:
1157 (latexrelease | fltrace | flafter) \the \@textmin}%
1158 (latexrelease | fltrace | flafter) \fl@trace{page-so-far: \the \@reqcolroom}%
1159 (latexrelease | fltrace | flafter)
1160 /trace)
1161 (latexrelease | fltrace | flafter) \ifdim \@textmin>\@reqcolroom

```

```

1162 <latexrelease | fltrace | flafter> \@reqcolroom \@textmin
1163 <*trace>
1164 <latexrelease | fltrace | flafter> \fl@trace{ORD? textmin being used}%
1165 </trace>
1166 <latexrelease | fltrace | flafter> \fi
1167 <latexrelease | fltrace | flafter> \advance \@reqcolroom \ht\@currbox
1168 <*trace>
1169 <latexrelease | fltrace | flafter> \fl@trace{float size =
1170 <latexrelease | fltrace | flafter> \the \ht \@currbox (addtocurcol)}%
1171 <latexrelease | fltrace | flafter> \fl@trace{colroom =
1172 <latexrelease | fltrace | flafter> \the \@colroom (addtocurcol)}%
1173 <latexrelease | fltrace | flafter> \fl@trace{reqcolroom =
1174 <latexrelease | fltrace | flafter> \the \@reqcolroom (addtocurcol)}%
1175 </trace>
1176 <latexrelease | fltrace | flafter> \ifdim \@colroom>\@reqcolroom
1177 <latexrelease | fltrace | flafter> \@flsetnum \@colnum
1178 <latexrelease | fltrace | flafter> \ifnum \@colnum>\z@
1179 <latexrelease | fltrace | flafter> \@bitor\@currtype\@deferlist
1180 <*trace>
1181 <latexrelease | fltrace | flafter> \fl@trace{deferlist:
1182 <latexrelease | fltrace | flafter> \@deferlist: (addtocurcol-before)}%
1183 </trace>
1184 <latexrelease | fltrace | flafter> \if@test
1185 <*trace>
1186 <latexrelease | fltrace | flafter> \fl@trace{type already on list:
1187 <latexrelease | fltrace | flafter> defer (addtocurcol)}%
1188 </trace>
1189 <latexrelease | fltrace | flafter> \else
1190 <latexrelease | fltrace | flafter> \@bitor\@currtype\@botlist
1191 <*trace>
1192 <latexrelease | fltrace | flafter> \fl@trace{botlist: \@botlist:
1193 <latexrelease | fltrace | flafter> (addtocurcol-before)}%
1194 </trace>
1195 <latexrelease | fltrace | flafter> \if@test
1196 <*trace>
1197 <latexrelease | fltrace | flafter> \fl@trace{type already on list:
1198 <latexrelease | fltrace | flafter> bot---sent to addtobot}%
1199 </trace>
1200 <latexrelease | fltrace | flafter> \@addtobot
1201 <latexrelease | fltrace | flafter> \else
1202 <*trace>
1203 <latexrelease | fltrace | flafter> \fl@trace{fpstype
1204 <latexrelease | fltrace | flafter> \ifodd \@tempcnta OK \else not \fi
1205 <latexrelease | fltrace | flafter> here: \the \@fpstype}%
1206 </trace>
1207 <latexrelease | fltrace | flafter> \ifodd \count\@currbox
1208 <latexrelease | fltrace | flafter> \advance \@reqcolroom \intextsep
1209 <latexrelease | fltrace | flafter> \ifdim \@colroom>\@reqcolroom
1210 <latexrelease | fltrace | flafter> \global \advance \@colnum \m@ne
1211 <latexrelease | fltrace | flafter> \global \advance
1212 <latexrelease | fltrace | flafter> \@textfloatsheight\ht\@currbox
This may sometimes give an overestimate.
1213 <latexrelease | fltrace | flafter> \global \advance
1214 <latexrelease | fltrace | flafter> \@textfloatsheight 2\intextsep

```

```

1215 (latexrelease | fltrace | flafter) \@cons \@midlist \@currbox
1216 (*trace)
1217 (latexrelease | fltrace | flafter) \fl@trace{***Success: here}%
1218 (latexrelease | fltrace | flafter) \fl@trace{textfloatsheight
1219 (latexrelease | fltrace | flafter) (after-here) =
1220 (latexrelease | fltrace | flafter) \the \@textfloatsheight}%
1221 (latexrelease | fltrace | flafter) \fl@trace{colnum (after-here) =
1222 (latexrelease | fltrace | flafter) \the \@colnum}%
1223 (/trace)

```

CHANGE TO \@addtocurcol:

\penalty\z@ changed to \penalty\interlinepenalty so \samepage works properly with figure and table environments. (Changed 23 Oct 86)

There is also an \addpenalty\interlinepenalty above.

Since in 2e \samepage is no longer supported, these could be removed.

Although it is best to use \addvspace in case two h floats come together, this makes other spacing more difficult to adjust; whereas if a user specifies two h floats together then they can more easily get the spacing correct by ad hoc commands.

It is necessary to adjust for the addition of \parskip here in case the float is added between paragraphs (i.e. when in vertical mode).

If the nobreak switch is true we need to reset it and clear \everypar since the float may not reset the flag and cannot reset the \everypar globally.

Typesetting starts here (we are in vertical mode).

```

1224 (latexrelease | fltrace | flafter) \if@nobreak
1225 (latexrelease | fltrace | flafter) \nobreak
1226 (latexrelease | fltrace | flafter) \@nobreakfalse
1227 (latexrelease | fltrace | flafter) \everypar{}%
1228 (latexrelease | fltrace | flafter) \else
1229 (latexrelease | fltrace | flafter) \addpenalty\interlinepenalty
1230 (latexrelease | fltrace | flafter) \fi
1231 (latexrelease | fltrace | flafter) \vskip \intextsep
1232 (latexrelease | fltrace | flafter) \box\@currbox
1233 (latexrelease | fltrace | flafter) \penalty\interlinepenalty
1234 (latexrelease | fltrace | flafter) \vskip\intextsep
1235 (latexrelease | fltrace | flafter) \ifnum\outputpenalty
1236 (latexrelease | fltrace | flafter) <-\@Mii \vskip
1237 (latexrelease | fltrace | flafter) -\parskip\fi

```

Typesetting ends here.

```

1238 (latexrelease | fltrace | flafter) \outputpenalty \z@
1239 (latexrelease | fltrace | flafter) \@inserttrue
1240 (*trace)
1241 (latexrelease | fltrace | flafter) \else
1242 (latexrelease | fltrace | flafter) \fl@trace{Fail---no room at 2nd test of colroom
1243 (latexrelease | fltrace | flafter) (addtocorcol \string\intextsep)}%
1244 (/trace)
1245 (latexrelease | fltrace | flafter) \fi
1246 (latexrelease | fltrace | flafter) \fi
1247 (latexrelease | fltrace | flafter) \if@insert
1248 (latexrelease | fltrace | flafter) \else

```

Next set of docstrip guards are a bit weird, essentially \@addtotoporbot ends up inside the kernel and the fltrace package and \@addtotoporbot shows up in the flafter package. Guess that could have been done a bit more obvious :-)

```

1249 {*2ekernel | fltrace)
1250 {*trace)
1251 {latexrelease | fltrace | flafter) \fl@trace{not here: sent to addtotoporbot}%
1252 {/trace)
1253 {latexrelease | fltrace | flafter) \@addtotoporbot
1254 {/2ekernel | fltrace)
1255 {*!2ekernel&!autoload&!fltrace)
1256 {*trace)
1257 {latexrelease | fltrace | flafter) \fl@trace{not here: sent to addtobot}%
1258 {/trace)
1259 {latexrelease | fltrace | flafter) \@addtobot
1260 {/!2ekernel&!autoload&!fltrace)
1261 {latexrelease | fltrace | flafter) \fi
1262 {latexrelease | fltrace | flafter) \fi
1263 {latexrelease | fltrace | flafter) \fi
1264 {*trace)
1265 {latexrelease | fltrace | flafter) \else
1266 {latexrelease | fltrace | flafter) \fl@trace{Fail: colnum = \the \@colnum:
1267 {latexrelease | fltrace | flafter) fpstype \the \@fpstype=ORD?}%
1268 {latexrelease | fltrace | flafter) \ifnum \@fpstype<\sist@n
1269 {latexrelease | fltrace | flafter) \fl@trace{ERROR: BANG float not successful
1270 {latexrelease | fltrace | flafter) (addtocurcol)}}%
1271 {latexrelease | fltrace | flafter) \fi
1272 {/trace)
1273 {latexrelease | fltrace | flafter) \fi
1274 {*trace)
1275 {latexrelease | fltrace | flafter) \else
1276 {latexrelease | fltrace | flafter) \fl@trace{Fail---no room: fl box ht:
1277 {latexrelease | fltrace | flafter) \the \ht \@currbox (addtocurcol)}}%
1278 {/trace)
1279 {latexrelease | fltrace | flafter) \fi
1280 {latexrelease | fltrace | flafter) \fi
1281 {latexrelease | fltrace | flafter) \fi
1282 {latexrelease | fltrace | flafter) \if@insert
1283 {latexrelease | fltrace | flafter) \else
1284 {latexrelease | fltrace | flafter) \@resethfps
1285 {*trace)
1286 {latexrelease | fltrace | flafter) \fl@trace{put on deferlist (addtocurcol)}}%
1287 {/trace)
1288 {latexrelease | fltrace | flafter) \@cons\@deferlist\@currbox
1289 {*trace)
1290 {latexrelease | fltrace | flafter) \fl@trace{deferlist: \@deferlist:
1291 {latexrelease | fltrace | flafter) (addtocurcol-after)}}%
1292 {/trace)
1293 {latexrelease | fltrace | flafter) \fi
1294 {latexrelease | fltrace | flafter) }%
1295 {latexrelease | fltrace | flafter)\EndIncludeInRelease

```

\@addtonextcol Lots of changes.

```

1296 {latexrelease | fltrace)\IncludeInRelease{2015/01/01}
1297 {latexrelease | fltrace) {\@addtonextcol}{float order in 2-column}%
1298 {*2ekernel | fltrace)
1299 \def\@addtonextcol{%
1300 \begingroup

```

```

1301 (*trace)
1302 \fl@trace{***Start addtonextcol}%
1303 </trace>
1304 \@insertfalse
1305 \@setfloattypecounts
1306 \ifnum \@fpstype=8
1307 (*trace)
1308 \fl@trace{fpstype not curcol: \the \@fpstype = 8?}%
1309 </trace>
1310 \else
1311 \ifnum \@fpstype=24
1312 (*trace)
1313 \fl@trace{fpstype not curcol: \the \@fpstype = 24?}%
1314 </trace>
1315 \else
1316 \@flsettextmin
1317 (*trace)
1318 \fl@trace{text-so-far: Opt (top of col)}%
1319 </trace>
1320 \@reqcolroom \ht\@currbox
1321 (*trace)
1322 \fl@trace{float size: \the \@reqcolroom (addtonextcol)}%
1323 </trace>
1324 \advance \@reqcolroom \@textmin
1325 (*trace)
1326 \fl@trace{colroom = \the \@colroom (addtonextcol)}%
1327 \fl@trace{reqcolroom = \the \@reqcolroom (addtonextcol)}%
1328 </trace>
1329 \ifdim \@colroom>\@reqcolroom
1330 \@flsetnum \@colnum
1331 \ifnum\@colnum>\z@
1332 \@bitor\@currtype\@deferlist
1333 (*trace)
1334 \fl@trace{deferlist: \@deferlist: (addtonextcol-before)}%
1335 </trace>
1336 \@testwrongwidth\@currbox
1337 \if@test
1338 (*trace)
1339 \fl@trace{type already on list: defer (addtonextcol)}%
1340 </trace>
1341 \else
1342 (*trace)
1343 \fl@trace{sent to addtotoporbot (addtonextcol)}%
1344 </trace>
1345 \@addtotoporbot
1346 \fi
1347 \fi
1348 (*trace)
1349 \else
1350 \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
1351 (addtonextcol)}%
1352 </trace>
1353 \fi

```

```

1354 \fi
1355 \fi
1356 \if@insert
1357 \else
1358 (*trace)
1359 \fl@trace{put back on deferlist (addtonextcol)}%
1360 </trace>
1361 \@cons\@deferlist\@currbox
1362 (*trace)
1363 \fl@trace{deferlist: \@deferlist: (addtonextcol-after)}%
1364 </trace>
1365 \fi
1366 (*trace)
1367 \fl@trace{End of addtonextcol -- locally counts:}%
1368 \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1369 </trace>
1370 \endgroup
1371 (*trace)
1372 \fl@trace{End of addtonextcol -- globally counts:}%
1373 \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
1374 </trace>
1375 }%
1376 </2ekernel | fltrace>
1377 <latexrelease | fltrace>\EndIncludeInRelease
1378 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
1379 <latexrelease | fltrace> {\@addfloatcol}{float order in 2-column}%
1380 <latexrelease | fltrace>\def\@addtonextcol{%
1381 <latexrelease | fltrace> \begingroup
1382 (*trace)
1383 <latexrelease | fltrace> \fl@trace{***Start addtonextcol}%
1384 </trace>
1385 <latexrelease | fltrace> \@insertfalse
1386 <latexrelease | fltrace> \@setfloattypecounts
1387 <latexrelease | fltrace> \ifnum \@fpstype=8
1388 (*trace)
1389 <latexrelease | fltrace> \fl@trace{fpstype not curcol:
1390 <latexrelease | fltrace> \the \@fpstype = 8?}%
1391 </trace>
1392 <latexrelease | fltrace> \else
1393 <latexrelease | fltrace> \ifnum \@fpstype=24
1394 (*trace)
1395 <latexrelease | fltrace> \fl@trace{fpstype not curcol:
1396 <latexrelease | fltrace> \the \@fpstype = 24?}%
1397 </trace>
1398 <latexrelease | fltrace> \else
1399 <latexrelease | fltrace> \@flsettextmin
1400 (*trace)
1401 <latexrelease | fltrace> \fl@trace{text-so-far: Opt (top of col)}%
1402 </trace>
1403 <latexrelease | fltrace> \@reqcolroom \ht\@currbox
1404 (*trace)
1405 <latexrelease | fltrace> \fl@trace{float size:
1406 <latexrelease | fltrace> \the \@reqcolroom (addtonextcol)}%
1407 <latexrelease | fltrace>

```

```

1408 </trace>
1409 <latexrelease | fltrace> \advance \@reqcolroom \@textmin
1410 <*trace>
1411 <latexrelease | fltrace> \fl@trace{colroom =
1412 <latexrelease | fltrace> \the \@colroom (addtonextcol)}%
1413 <latexrelease | fltrace> \fl@trace{reqcolroom =
1414 <latexrelease | fltrace> \the \@reqcolroom (addtonextcol)}%
1415 </trace>
1416 <latexrelease | fltrace> \ifdim \@colroom>\@reqcolroom
1417 <latexrelease | fltrace> \@flsetnum \@colnum
1418 <latexrelease | fltrace> \ifnum\@colnum>\z@
1419 <latexrelease | fltrace> \@bitor\@currtype\@deferlist
1420 <*trace>
1421 <latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1422 <latexrelease | fltrace> (addtonextcol-before)}%
1423 </trace>
1424 <latexrelease | fltrace> \if@test
1425 <*trace>
1426 <latexrelease | fltrace> \fl@trace{type already on list:
1427 <latexrelease | fltrace> defer (addtonextcol)}%
1428 </trace>
1429 <latexrelease | fltrace> \else
1430 <*trace>
1431 <latexrelease | fltrace> \fl@trace{sent to addtotoporbot
1432 <latexrelease | fltrace> (addtonextcol)}%
1433 </trace>
1434 <latexrelease | fltrace> \@addtotoporbot
1435 <latexrelease | fltrace> \fi
1436 <latexrelease | fltrace> \fi
1437 <*trace>
1438 <latexrelease | fltrace> \else
1439 <latexrelease | fltrace> \fl@trace{Fail---no room: fl box ht:
1440 <latexrelease | fltrace> \the \ht \@currbox (addtonextcol)}%
1441 </trace>
1442 <latexrelease | fltrace> \fi
1443 <latexrelease | fltrace> \fi
1444 <latexrelease | fltrace> \fi
1445 <latexrelease | fltrace> \if@insert
1446 <latexrelease | fltrace> \else
1447 <*trace>
1448 <latexrelease | fltrace> \fl@trace{put back on deferlist
1449 <latexrelease | fltrace> (addtonextcol)}%
1450 </trace>
1451 <latexrelease | fltrace> \@cons\@deferlist\@currbox
1452 <*trace>
1453 <latexrelease | fltrace> \fl@trace{deferlist: \@deferlist:
1454 <latexrelease | fltrace> (addtonextcol-after)}%
1455 </trace>
1456 <latexrelease | fltrace> \fi
1457 <*trace>
1458 <latexrelease | fltrace> \fl@trace{End of addtonextcol --
1459 <latexrelease | fltrace> locally counts:}%
1460 <latexrelease | fltrace> \fl@trace{col: \the \@colnum.
1461 <latexrelease | fltrace> top: \the \@topnum. bot: \the \@botnum.}%

```

```

1462 </trace>
1463 <latexrelease | fltrace> \endgroup
1464 <*trace>
1465 <latexrelease | fltrace> \fl@trace{End of addtonextcol --
1466 <latexrelease | fltrace> globally counts:}%
1467 <latexrelease | fltrace> \fl@trace{col: \the \@colnum.
1468 <latexrelease | fltrace> top: \the \@topnum. bot: \the \@botnum.}%
1469 </trace>
1470 <latexrelease | fltrace>}%
1471 <latexrelease | fltrace>\EndIncludeInRelease

```

\@addtodblcol Lots of changes.

```

1472 <latexrelease | fltrace>\IncludeInRelease{2015/01/01}%
1473 <latexrelease | fltrace> {\@addtodblcol}{float order in 2-column}%
1474 <*2ekernel | latexrelease | fltrace>
1475 \def\@addtodblcol{%
1476 \begingroup
1477 <*trace>
1478 \fl@trace{***Start addtodblcol}%
1479 </trace>
1480 \@insertfalse
1481 \@setfloattyperecounts
1482 \@getfpsbit \tw@
1483 <*trace>
1484 \fl@trace{fpstype \ifodd \@tempcnta OK \else not \fi dbltop:
1485 \the \@fpstype}%
1486 </trace>
1487 \ifodd\@tempcnta
1488 \@flsetnum \@dbltopnum
1489 \ifnum \@dbltopnum>\z@
1490 \@tempwafalse
1491 \ifdim \@dbltoproom>\ht\@currbox
1492 \@tempwatruer
1493 <*trace>
1494 \fl@trace{Space OK: \@dbltoproom =
1495 \the \@dbltoproom > \the \ht \@currbox
1496 (dbltoproom)}%
1497 </trace>
1498 \else
1499 <*trace>
1500 \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
1501 </trace>
1502 \ifnum \@fpstype<\sist@n
1503 <*trace>
1504 \fl@trace{BANG float ignoring \@dbltoproom}%
1505 \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1506 Ht float: \the \ht \@currbox-BANG}%
1507 </trace>

```

Need to check that there is room on the page, using the local value of \@textmin to make the necessary adjustment to \@dbltoproom.

```

1508 \advance \@dbltoproom \@textmin
1509 <*trace>
1510 \fl@trace{Local value of texmin: \the\@textmin}%

```



```

1511 \fl@trace{\@spaces space on page = \the \@dbltoproom.
1512 Ht float: \the \ht \@currbox-BANG}%
1513 </trace>
1514 \ifdim \@dbltoproom>\ht\@currbox
1515 \@tempwattrue
1516 (*trace)
1517 \fl@trace{Space OK BANG: space on page =
1518 \the \@dbltoproom > \the \ht \@currbox}%
1519 \else
1520 \fl@trace{fpstype: \the \@fpstype}%
1521 \fl@trace{Fail---no room dbltoproom-BANG?:}%
1522 \fl@trace{\@spaces space on page = \the \@dbltoproom.
1523 Ht float: \the \ht \@currbox}%
1524 </trace>
1525 \fi
1526 \advance \@dbltoproom -\@textmin
1527 (*trace)
1528 \else
1529 \fl@trace{fpstype: \the \@fpstype}%
1530 \fl@trace{Fail---no room dbltoproom-ORD?:}%
1531 \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
1532 Ht float: \the \ht \@currbox}%
1533 </trace>
1534 \fi
1535 \fi
1536 \if@tempswa
1537 \@bitor \@currtype \@deferlist
1538 (*trace)
1539 \fl@trace{(dbl)deferlist: \@deferlist: (before)}%
1540 </trace>
1541 not in fixfloats?
1542 \@testwrongwidth\@currbox
1543 (*trace)
1544 \if@test
1545 \fl@trace{type already on list: (dbl)defer}%
1546 \else
1547 \@tempdima -\ht\@currbox
1548 \advance\@tempdima
1549 -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
1550 \dblfloatsep \fi
1551 \global \advance \@dbltoproom \@tempdima
1552 \global \advance \@colht \@tempdima
1553 \global \advance \@dbltopnum \m@ne
1554 \@cons \@dbltoplist \@currbox
1555 (*trace)
1556 \fl@trace{dbltopnum (after) = \the \@dbltopnum}%
1557 \fl@trace{***Success: dbltop}%
1558 </trace>
1559 \@inserttrue
1560 \fi
1561 \fi
1562 (*trace)

```

```

1563 \else
1564 \fl@trace{Fail: dbltopnum = \the \@dbltopnum: fpstype
1565 \the \@fpstype=ORD?}%
1566 \ifnum \@fpstype<\sixt@n
1567 \fl@trace{ERROR: !t float not successful (addtodblcol)}%
1568 \fi
1569 </trace>
1570 \fi
1571 \fi
1572 \if@insert
1573 \else
1574 (*trace)
1575 \fl@trace{put on deferlist}%
1576 </trace>
1577 \@cons\@deferlist\@currbox
1578 (*trace)
1579 \fl@trace{(dbl)deferlist: \@deferlist: (after)}%
1580 </trace>
1581 \fi
1582 (*trace)
1583 \fl@trace{End of addtodblcol -- locally count:}%
1584 \fl@trace{ dbltop: \the \@dbltopnum.}%
1585 </trace>
1586 \endgroup
1587 (*trace)
1588 \fl@trace{End of addtodblcol -- globally count:}%
1589 \fl@trace{dbltop: \the \@dbltopnum.}%
1590 </trace>
1591 }%
1592 </2kernel | latexrelease | fltrace>
1593 <latexrelease | fltrace>\EndIncludeInRelease
1594 <latexrelease | fltrace>\IncludeInRelease{0000/00/00}%
1595 <latexrelease | fltrace> {\@addtodblcol}{float order in 2-column}%
1596 <latexrelease | fltrace>\def\@addtodblcol{%
1597 <latexrelease | fltrace> \begingroup
1598 (*trace)
1599 <latexrelease | fltrace> \fl@trace{***Start addtodblcol}%
1600 </trace>
1601 <latexrelease | fltrace> \@insertfalse
1602 <latexrelease | fltrace> \@setfloattypecounts
1603 <latexrelease | fltrace> \@getfpsbit \tw@
1604 (*trace)
1605 <latexrelease | fltrace> \fl@trace{fpstype \ifodd \@tempcnta OK
1606 <latexrelease | fltrace> \else not \fi dbltop: \the \@fpstype}%
1607 </trace>
1608 <latexrelease | fltrace> \ifodd\@tempcnta
1609 <latexrelease | fltrace> \@flsetnum \@dbltopnum
1610 <latexrelease | fltrace> \ifnum \@dbltopnum>\z@
1611 <latexrelease | fltrace> \@tempswafalse
1612 <latexrelease | fltrace> \ifdim \@dbltoproom>\ht\@currbox
1613 <latexrelease | fltrace> \@tempswatrue
1614 (*trace)
1615 <latexrelease | fltrace> \fl@trace{Space OK: \@dbltoproom =
1616 <latexrelease | fltrace> \the \@dbltoproom > \the \ht \@currbox

```

```

1617 \latexrelease | fltrace) (dbltoproom}}%
1618 \trace)
1619 \latexrelease | fltrace) \else
1620 (*trace)
1621 \latexrelease | fltrace) \fl@trace{fpstype: \the \@fpstype (addtodblcol}}%
1622 \trace)
1623 \latexrelease | fltrace) \ifnum \@fpstype<\sift@@n
1624 (*trace)
1625 \latexrelease | fltrace) \fl@trace{BANG float ignoring \dbltoproom}}%
1626 \latexrelease | fltrace) \fl@trace{\@spaces \dbltoproom =
1627 \latexrelease | fltrace) \the \dbltoproom.
1628 \latexrelease | fltrace) Ht float: \the \ht \@currbox-BANG}}%
1629 \trace)

```

Need to check that there is room on the page, using the local value of `\@textmin` to make the necessary adjustment to `\dbltoproom`.

```

1630 \latexrelease | fltrace) \advance \dbltoproom \@textmin
1631 (*trace)
1632 \latexrelease | fltrace) \fl@trace{Local value of texmin: \the\@textmin}}%
1633 \latexrelease | fltrace) \fl@trace{\@spaces space on page =
1634 \latexrelease | fltrace) \the \dbltoproom.
1635 \latexrelease | fltrace) Ht float: \the \ht \@currbox-BANG}}%
1636 \trace)
1637 \latexrelease | fltrace) \ifdim \dbltoproom>\ht\@currbox
1638 \latexrelease | fltrace) \@tempswatruel
1639 (*trace)
1640 \latexrelease | fltrace) \fl@trace{Space OK BANG: space on page =
1641 \latexrelease | fltrace) \the\dbltoproom > \the\ht\@currbox}}%
1642 \latexrelease | fltrace) \else
1643 \latexrelease | fltrace) \fl@trace{fpstype: \the \@fpstype}}%
1644 \latexrelease | fltrace) \fl@trace{Fail---no room dbltoproom-BANG?:}}%
1645 \latexrelease | fltrace) \fl@trace{\@spaces space on page =
1646 \latexrelease | fltrace) \the \dbltoproom.
1647 \latexrelease | fltrace) Ht float: \the \ht \@currbox}}%
1648 \trace)
1649 \latexrelease | fltrace) \fi
1650 \latexrelease | fltrace) \advance \dbltoproom -\@textmin
1651 (*trace)
1652 \latexrelease | fltrace) \else
1653 \latexrelease | fltrace) \fl@trace{fpstype: \the \@fpstype}}%
1654 \latexrelease | fltrace) \fl@trace{Fail---no room dbltoproom-ORD?:}}%
1655 \latexrelease | fltrace) \fl@trace{\@spaces \dbltoproom =
1656 \latexrelease | fltrace) \the \dbltoproom.
1657 \latexrelease | fltrace) Ht float: \the \ht \@currbox}}%
1658 \trace)
1659 \latexrelease | fltrace) \fi
1660 \latexrelease | fltrace) \fi
1661 \latexrelease | fltrace) \if@tempswa
1662 \latexrelease | fltrace) \@bitor \@currtype \@dbldeferlist
1663 (*trace)
1664 \latexrelease | fltrace) \fl@trace{dbldeferlist:
1665 \latexrelease | fltrace) \dbldeferlist: (before)}}%
1666 \trace)
1667 \latexrelease | fltrace) \if@test

```

```

1668 (*trace)
1669 (latexrelease | fltrace) \fl@trace{type already on list: dbldefer}%
1670 (/trace)
1671 (latexrelease | fltrace) \else
1672 (latexrelease | fltrace) \@tempdima -\ht\@currbox
1673 (latexrelease | fltrace) \advance\@tempdima
1674 (latexrelease | fltrace) -\ifx \@dbltoplist\@empty
1675 (latexrelease | fltrace) \dbltextfloatsep
1676 (latexrelease | fltrace) \else \dblfloatsep \fi
1677 (latexrelease | fltrace) \global \advance \@dbltoproom \@tempdima
1678 (latexrelease | fltrace) \global \advance \@colht \@tempdima
1679 (latexrelease | fltrace) \global \advance \@dbltopnum \m@ne
1680 (latexrelease | fltrace) \@cons \@dbltoplist \@currbox
1681 (*trace)
1682 (latexrelease | fltrace) \fl@trace{dbltopnum (after) =
1683 (latexrelease | fltrace) \the \@dbltopnum}%
1684 (latexrelease | fltrace) \fl@trace{***Success: dbltop}%
1685 (/trace)
1686 (latexrelease | fltrace) \@inserttrue
1687 (latexrelease | fltrace) \fi
1688 (latexrelease | fltrace) \fi
1689 (*trace)
1690 (latexrelease | fltrace) \else
1691 (latexrelease | fltrace) \fl@trace{Fail: dbltopnum = \the \@dbltopnum:
1692 (latexrelease | fltrace) fpstype \the \@fpstype=ORD?}%
1693 (latexrelease | fltrace) \ifnum \@fpstype<\sift@n
1694 (latexrelease | fltrace) \fl@trace{ERROR: !t float not successful
1695 (latexrelease | fltrace) (addtodblcol)}%
1696 (latexrelease | fltrace) \fi
1697 (/trace)
1698 (latexrelease | fltrace) \fi
1699 (latexrelease | fltrace) \fi
1700 (latexrelease | fltrace) \if@insert
1701 (latexrelease | fltrace) \else
1702 (*trace)
1703 (latexrelease | fltrace) \fl@trace{put on dbldeferlist}%
1704 (/trace)
1705 (latexrelease | fltrace) \@cons\@dbldeferlist\@currbox
1706 (*trace)
1707 (latexrelease | fltrace) \fl@trace{dbldeferlist: \@dbldeferlist: (after)}%
1708 (/trace)
1709 (latexrelease | fltrace) \fi
1710 (*trace)
1711 (latexrelease | fltrace) \fl@trace{End of addtodblcol -- locally count:}%
1712 (latexrelease | fltrace) \fl@trace{dbltop: \the \@dbltopnum.}%
1713 (/trace)
1714 (latexrelease | fltrace) \endgroup
1715 (*trace)
1716 (latexrelease | fltrace) \fl@trace{End of addtodblcol -- globally count:}%
1717 (latexrelease | fltrace) \fl@trace{dbltop: \the \@dbltopnum.}%
1718 (/trace)
1719 (latexrelease | fltrace) }%
1720 (latexrelease | fltrace) \EndIncludeInRelease

```

`\@addmarginpar`

```
1721 (*2kernel)
1722 \def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
1723 \@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
1724 \if@twocolumn
1725 \if@firstcolumn \@tempcnta\m@ne \fi
1726 \else
1727 \if@mparswitch
1728 \ifodd\c@page \else\@tempcnta\m@ne \fi
1729 \fi
1730 \if@reversemargin \@tempcnta -\@tempcnta \fi
1731 \fi
1732 \ifnum\@tempcnta <\z@ \global\setbox\@marbox\box\@currbox \fi
1733 \@tempdima\@mparbottom
1734 \advance\@tempdima -\@pageht
1735 \advance\@tempdima\ht\@marbox
1736 \ifdim\@tempdima >\z@
1737 \@latex@warning@no@line {Marginpar on page \thepage\space moved}%
1738 \else
1739 \@tempdima\z@
1740 \fi
1741 \global\@mparbottom\@pageht
1742 \global\advance\@mparbottom\@tempdima
1743 \global\advance\@mparbottom\dp\@marbox
1744 \global\advance\@mparbottom\marginparpush
1745 \advance\@tempdima -\ht\@marbox
```

Putting box movement inside the ‘marbox’:

```
1746 \global\setbox \@marbox
1747 \vbox {\vskip \@tempdima
1748 \box \@marbox}%
1749 \global \ht\@marbox \z@
1750 \global \dp\@marbox \z@
```

Sticking (rather than gluing:-) the ‘marbox’ to the line above, changed vskip to kern:

```
1751 \kern -\@pagedp
1752 \nointerlineskip
1753 \hb@xt@\columnwidth
1754 {\ifnum \@tempcnta >\z@
1755 \hskip\columnwidth \hskip\marginparsep
1756 \else
1757 \hskip -\marginparsep \hskip -\marginparwidth
1758 \fi
1759 \box\@marbox \hss}%
```

For this reason the following code can vanish:

```
\nobreak %% No longer needed. CAR92/12
\vskip -\@tempdima %% No longer needed. CAR92/12

1760 \nointerlineskip
1761 \hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}
```

### 64.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for L<sup>A</sup>T<sub>E</sub>X2e.

```
\enlargethispage{<dim>}
```

Adds <dim> to the height of the current column only. On the printed page the bottom of this column is extended downwards by exactly <dim> without having any effect on the placement of the footer; this may result in an overprinting.

```
\enlargethispage*{<dim>}
```

Similar to `\enlargethispage` but it tries to squeeze the column to be printed in as small a space as possible, ie it uses any shrinkability in the column. If the column was not explicitly broken (e.g. with `\pagebreak`) this may result in an overfull box message but except for this it will come out as expected (if you know what to expect).

The star form of this command is dedicated to Leslie Lamport, the other we need for ourselves (FMi, CAR).

These commands may well have unwanted effects if used soon before a `\clearpage`: please give keep them clear of such places.

`\@kludgeins` The insert which makes T<sub>E</sub>X do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```
1762 \newinsert \@kludgeins
1763 \global\dimen\@kludgeins \maxdimen
1764 \global\count\@kludgeins 1000
```

`\enlargethispage` The user command.

```
\enlargethispage* 1765 \gdef \enlargethispage {%
1766 \ifstar
1767 {%
1768 (*trace)
1769 \fl@trace{Enlarging page height * }%
1770 </trace>
1771 \@enlargepage{\hbox{\kern\p@}}%
1772 {%
1773 (*trace)
1774 \fl@trace{Enlarging page height exactly---}%
1775 </trace>
1776 \@enlargepage\@empty}%
1777 }
```

`\@enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```
1778 \gdef\@enlargepage#1#2{%
1779 (*trace)
1780 \fl@trace{\@spaces\@spaces by #2}%
1781 </trace>
1782 \@tempkipa#2\relax
1783 \ifdim \@tempkipa>.5\maxdimen
```

```

1784 \@latexerr{Suggested\space extra\space height\space
1785 (\the\@tempskipa)\space dangerously\space
1786 large}\@eha
1787 \else
1788 \ifdim \vsize<.5\maxdimen
1789 (*trace)
1790 \fl@trace {Kludgeins added--pagegoal before: \the\pagegoal}%
1791 </trace>
1792 \@bsphack
1793 \insert\@kludgeins{#1\vskip-\@tempskipa}%
1794 \@esphack

```

This next bit is for tracing only:

```

1795 (*trace)
1796 \ifvmode \par
1797 \fl@trace {Kludgeins added--pagegoal after: \the\pagegoal}%
1798 \fi
1799 </trace>
1800 \else
1801 \@latexerr{Page\space height\space already\space
1802 too\space large}\@eha
1803 \fi
1804 \fi
1805 }
1806 </2kernel>

```

### 64.1.2 Float control

This part implements controllable floats and other changes to the float mechanism.

It provides, at the document level, the following command for inclusion in L<sup>A</sup>T<sub>E</sub>X<sub>2</sub>ε.

```
\suppressfloats
```

This suppresses all further floats on the current page.

With an optional argument it suppresses only floats only in certain positions on the current page.

[t] suppresses only floats at the top of the page [b] suppresses only floats at the bottom of the page

It also enables the use of an extra specifier, !, in the location optional argument of a float. If this is present then, just for this particular float, whenever it is processed by the float mechanism the followinghg are ignored:

- all restrictions on the number of floats which can appear;
- all explicit restrictions on the amount of space which should (not) be occupied by floats and/or text.

The mechanism will still attempt to ensure that pages are not overfull.

These specifiers override, for the single float, the suppression commands described above.

In its current form, it also supplies a reasonably exhaustive, and somewhat baroque, means of tracing some aspects of the float mechanism.

More tracing.

```

\fl@trace Set-up tracing for floats independent of other tracing as it produces mega-output.
\tracefloatsoff Default is no tracing.
\tracefloats 1807 (*fltrace)
\fl@traceval 1808 \def \fl@tracemessage #1{\let\@elt\@empty\typeout{LaTeX2e: #1}}
\tracefloatvals 1809 \def \tracefloats{\let \fl@trace \fl@tracemessage}
\fl@tracemessage 1810 \def \tracefloatsoff {\let \fl@trace \@gobble}
1811 \tracefloatsoff
1812 \def \fl@traceval #1{\fl@trace{\string #1 = \the #1}}
1813 \IncludeInRelease{2015/01/01}{\tracefloatvals}%
1814 {trace float vals}%
1815 \def \tracefloatvals{%

As \@dblfloatplacement sets \f@depth it needs to be run inside a group, otherwise the float placement will test for the wrong value.8

1816 \begingroup
1817 \@dblfloatplacement
1818 \@floatplacement
1819 \fl@trace{***Float placement parameters:}%
1820 \fl@traceval\@colnum
1821 \fl@traceval\@colroom
1822 \fl@traceval\@topnum
1823 \fl@traceval\@toproom
1824 \fl@traceval\@botnum
1825 \fl@traceval\@botroom
1826 \fl@traceval\@fpmin
1827 \fl@trace{\string\textfraction = \textfraction}%
1828 \fl@traceval\@dbltopnum
1829 \fl@traceval\@dbltoproom
1830 \fl@trace{\string\textfraction = \textfraction}%
1831 \fl@trace{toplist: \@toplist}%
1832 \fl@trace{botlist: \@botlist}%
1833 \fl@trace{midlist: \@midlist}%
1834 \fl@trace{deferlist: \@deferlist}%
1835 \fl@trace{dbltoplist: \@dbltoplist}%
1836 %Fmi \fl@trace{dbldeferlist: \@dbldeferlist}%
1837 \endgroup
1838 }
1839 \EndIncludeInRelease
1840 \IncludeInRelease{0000/00/00}{\tracefloatvals}%
1841 {trace float vals}%
1842 \def \tracefloatvals{%
1843 \begingroup
1844 \@dblfloatplacement
1845 \@floatplacement
1846 \fl@trace{***Float placement parameters:}%
1847 \fl@traceval\@colnum
1848 \fl@traceval\@colroom
1849 \fl@traceval\@topnum

```

---

<sup>8</sup>This is a somewhat questionable design.



```

1850 \fl@traceval\@toproom
1851 \fl@traceval\@botnum
1852 \fl@traceval\@botroom
1853 \fl@traceval\@fpmin
1854 \fl@trace{\string\textfraction = \textfraction}%
1855 \fl@traceval\@dbltopnum
1856 \fl@traceval\@dbltoproom
1857 \fl@trace{\string\textfraction = \textfraction}%
1858 \fl@trace{toplist: \@toplist}%
1859 \fl@trace{botlist: \@botlist}%
1860 \fl@trace{midlist: \@midlist}%
1861 \fl@trace{deferlist: \@deferlist}%
1862 \fl@trace{dbltoplist: \@dbltoplist}%
1863 % next line only in old releases
1864 \fl@trace{dbldeferlist: \@dbldeferlist}%
1865 \endgroup
1866 }
1867 \EndIncludeInRelease

```

We need to make sure that `fltrace` comes before `flafter` to make the tracing work.

```

1868 \@ifpackageloaded{flafter}
1869 {\PackageWarningNoLine
1870 {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
1871 Attempting to recover by reloading 'flafter'}}%

```

Hide the fact that `flafter` was already loaded and then request it anew.

```

1872 \expandafter\let\csname ver@flafter.sty\endcsname\relax
1873 \def\reserved@a#1{%
1874 \expandafter\let\csname\string#1+flafter+IIR\endcsname\relax}%
1875 \reserved@a\@addtocurcol
1876 \reserved@a\@addtonextcol
1877 \RequirePackage{flafter}}{}
1878 </fltrace>

```

As the code for `flafter` will contain tracing calls so that it works in conjunction with `fltrace` we need to provide a dummy definition for `\fl@trace` in that package.

```

1879 (*flafter)
1880 \providecommand\fl@trace[1]{}
1881 </flafter>

```

`\suppressfloats` Float suppression commands: these set the relevant counter globally to zero. Thus `\@flstop` they are overridden for a particular float by an `!` specifier.

```

1882 (*2kernel)
1883 \def \suppressfloats {%
1884 \@ifnextchar [%
1885 \@flstop
1886 {\global \@colnum \z@}%
1887 }

```

Maybe this should be a loop over `#1`?

```

1888 \def \@flstop [#1] {%
1889 \if t#1%
1890 \global \@topnum \z@

```

```

1891 \fi
1892 \if b#1%
1893 \global \@botnum \z@
1894 \fi
1895 }

```

Manipulation of float placement and type; both their strings and the corresponding count registers.

`\@fpstype` First a new count register to go with `\@currtype`.  
`\@reqcolroom` Then a new skip register, for information needed to remove the `\@maxsep` conservatism: it is possible that this could use a temporary register.

Finally a dimension register to hold the total height of in-text floats on the current page. This is needed to implement a major change in the functionality of `\@addtocurcol` which is, nevertheless, a bug fix. It is not local and therefore cannot be a temporary register.

```

1896 \newcount \@fpstype
1897 \newdimen \@reqcolroom
1898 \newdimen \@textfloatsheight
1899 \if2kernel

```

`\@fpsadddefault` Adds the default placement to what is already there.  
Should not need to change this, but could do it as follows:

```

\def \@fpsadddefault {%
 \@temptokena \expandafter\expandafter\expandafter
 {\csname fps@\@capttype \endcsname}%
 \edef \reserved@a {\the\@temptokena}%
 \@onelevel@sanitize \reserved@a
 \edef \@fps {\@fps\reserved@a}%
}

```

```

1900 \if2kernel|ftrace)
1901 \def \@fpsadddefault {%
1902 (*trace)
1903 \fl@trace{fps changed from: \@fps}%
1904 \iftrace)
1905 \edef \@fps {\@fps\csname fps@\@capttype \endcsname}%
1906 \@latex@warning {%
1907 No positions in optional float specifier.\MessageBreak
1908 Default added (so using '\@fps')}%
1909 }

```

`\@setfloattypecounts` Sets counters `\@fpstype` and `\@currtype`.  
BANG == bit4 of `\count\@currbox = 0`.

```

1910 \def \@setfloattypecounts {%
1911 \@currtype \count\@currbox
1912 \@fpstype \count\@currbox
1913 \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
1914 \advance \@fpstype -\@currtype
1915 (*trace)
1916 \fl@trace{(mod 32) fpstype: \the \@fpstype}%
1917 \fl@trace{(mult of 32) currtype: \the \@currtype}%

```

```

1918 % Tracing only: but some should be changed into real errors/warnings?
1919 \ifnum \@fpstype<\sist@n
1920 \ifnum \@fpstype=\z@
1921 \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
1922 \fi
1923 \ifnum \@fpstype=\@ne
1924 \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 1?}%
1925 \fi
1926 \fl@trace{BANG float}%
1927 \else
1928 \ifnum \@fpstype=\sist@n
1929 \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
1930 \fi
1931 \ifnum \@fpstype=17
1932 \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 17?}%
1933 \fi
1934 \fl@trace{ORD float}%
1935 \fi
1936 </trace>
1937 }
1938 </2ekernel | fltrace>

```

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```

1939 (*2ekernel)
1940 \def \@getfpsbit {%
1941 \@boxfpsbit \@currbox
1942 }

```

`\@boxfpsbit` Used above.

```

1943 \def \@boxfpsbit #1#2{%
1944 \@tempcnta \count#1
1945 \divide \@tempcnta #2\relax
1946 }

```

`\@testfp` New definition of the float page test.

```

1947 \def \@testfp #1{%
1948 \@boxfpsbit #18\relax % Really ‘#1 8’ for human readers!
1949 \ifodd \@tempcnta
1950 \else
1951 \@testtrue
1952 \fi
1953 }

```

`\@setfpsbit` Sets required bit of `\@tempcnta` (to 1).

```

1954 \def \@setfpsbit #1{%
1955 \@tempcntb \@tempcnta
1956 \divide \@tempcntb #1\relax
1957 \ifodd \@tempcntb
1958 \else
1959 \advance \@tempcnta #1\relax
1960 \fi

```

```

1961 }
1962 </2ekernel)

```

`\@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`.

```

1963 (*2ekernel | fltrace)
1964 \def \@resethfps {%
1965 \let\reserved@a\@empty
1966 \ifnum \@fpstype=\@ne
1967 \def \reserved@a {!}%
1968 \@fpstype 17
1969 \fi
1970 \ifnum \@fpstype=17
1971 \global \advance \count\@currbox \tw@
1972 \@latex@warning@no@line {%
1973 '\reserved@a h' float specifier changed to '\reserved@a ht'}%
1974 (*trace)
1975 \fl@trace{%
1976 't' added to '\reserved@a h'- new Count: \the \count\@currbox}%
1977 </trace)
1978 \fi
1979 }

```

Special stuff for BANG floats.

`\@flsetnum` Ignores any zero float counter value in case BANG.

It uses a local assignment to the normally global counter: a bit naughty, perhaps?

These assignments are safe so long as the counter involved is only consulted once (i.e. only for the ‘bang float’) with the changed value. This is the case within `\@addtocurcol` because it is used only once within a call of the output routine (which forms a group).

For `\@addtonextcol` this is achieved by putting a group around its code; this is needed because it is called (by `\@startcolumn`) for each float which was on the deferlist. Almost identical considerations pertain to `\@addtodblcol`. There may be more efficient ways to handle this, but the group seems to be the simplest.

```

1980 \def \@flsetnum #1{%
1981 (*trace)
1982 \fl@trace{fpstype: \the \@fpstype (flsetnum \string#1)}%
1983 </trace)
1984 \ifnum \@fpstype<\sxt@n
1985 \ifnum #1=\z@
1986 (*trace)
1987 \fl@trace{BANG float resetting \string#1 to 1}%
1988 </trace)
1989 #1\@ne
1990 \fi
1991 \fi
1992 (*trace)
1993 \fl@trace{#1 (before) = \the #1}%
1994 </trace)

```

1995 }

`\@flsettextmin` This ignores `\textfraction` space restriction in case BANG.

```
1996 \def \@flsettextmin {%
1997 (*trace)
1998 \fl@trace{fpstype: \the \@fpstype (flsettextmin)}%
1999 </trace>
2000 \ifnum \@fpstype<\sixt@n
2001 (*trace)
2002 \fl@trace{BANG ignoring textmin}%
2003 </trace>
2004 \@textmin \z@
2005 \else
2006 \@textmin \textfraction\colht
2007 (*trace)
2008 \fl@trace{ORD textmin = \the \@textmin}%
2009 </trace>
2010 \fi
2011 }
```

`\@flcheckspace` This ignores space restriction in case BANG; this is still slightly conservative since it does not allow for the fact that, if there is no text in the column then `\textfloatsep` is not needed. Sets `@tempwa` true if there is room for `\@currbox`.

```
2012 \def \@flcheckspace #1#2{%
2013 \advance \@reqcolroom
2014 \ifx #2\empty \textfloatsep \else \floatsep \fi
2015 (*trace)
2016 \fl@trace{colroom = \the \@colroom
2017 (flcheckspace \string#1 \string#2)}%
2018 \fl@trace{reqcolroom = \the \@reqcolroom
2019 (flcheckspace \string#1 \string#2)}%
2020 </trace>
2021 \ifdim \@colroom>\@reqcolroom
2022 \ifdim #1>\ht\@currbox
2023 \@tempwattrue
2024 (*trace)
2025 \fl@trace{Space OK: #1 = \the #1 > \the \ht \@currbox
2026 (flcheckspace \string#1 \string#2)}%
2027 </trace>
2028 \else
2029 (*trace)
2030 \fl@trace{fpstype: \the \@fpstype
2031 (flcheckspace \string#1 \string#2)}%
2032 </trace>
2033 \ifnum \@fpstype<\sixt@n
2034 (*trace)
2035 \fl@trace{BANG float ignoring #1
2036 (flcheckspace \string#1 \string#2):}%
2037 \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2038 BANG}%
2039 </trace>
2040 \@tempwattrue
2041 (*trace)
```

```

2042 \else
2043 \fl@trace{Fail---no room (flcheckspace \string#1 \string#2)
2044 (fpstype \the \@fpstype=ORD?):}%
2045 \fl@trace{\@spaces #1 = \the #1. Ht float: \the \ht \@currbox
2046 ORD?}%
2047 </trace>
2048 \fi
2049 \fi
2050 (*trace)
2051 \else
2052 \fl@trace{Fail---no room at 2nd test of colroom
2053 (flcheckspace \string#1 \string#2)}%
2054 </trace>
2055 \fi
2056 }
2057 </2ekernel | fltrace)

```

`\@flupdates` This updates everything when a float is placed.

```

2058 (*2ekernel)
2059 \def \@flupdates #1#2#3{%
2060 \global \advance #1\m@ne
2061 \global \advance \@colnum \m@ne
2062 \@tempdima -\ht\@currbox
2063 \advance \@tempdima
2064 -\ifx #3\@empty \textfloatsep \else \floatsep \fi
2065 \global \advance #2\@tempdima
2066 \global \advance \@colroom \@tempdima
2067 \@cons #3\@currbox
2068 }
2069 </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.

```
2070 (*2ekernel | fltrace)
2071 \def\@makefcolumn #1{%
2072 \begingroup
2073 \@fpmin \z@
2074 \let \@testfp \@gobble
```



```

2075 \@tryfcolumn #1%
2076 \endgroup
2077 (*trace)
2078 \if@colmade
2079 \fl@trace{PAGE: in \string\clearpage
2080 \if@twocolumn ---twocolumn\fi---}%
2081 \fl@trace{----- float column/page completed from \string#1}%
2082 \fi
2083 \trace)
2084 }

```

This will line up the last baselines in the two columns provided they are constructed in the normal way: i.e. ending in a skip of minus the original depth, with `\@textbottom` adding nothing.

Thus again it is essential for `\@textbottom` to have depth 0pt.

```

2085 \2kernel | fltrace)
2086 (latexrelease | fltrace)\IncludeInRelease{2015/01/01}%
2087 (latexrelease | fltrace) {\@outputdblcol}{2 column marks}%
2088 (*2kernel | fltrace | latexrelease)

```

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```

2089 \def\@outputdblcol{%
2090 \if@firstcolumn
2091 \global\@firstcolumnfalse

```

Save the left column

```

2092 \global\setbox\@leftcolumn\copy\@outputbox
2093 (fltrace) \fl@trace{PAGE: first column boxed}%

```

Remember the marks from the first column

```

2094 \splitmaxdepth\maxdimen
2095 \vbadness\maxdimen

```

In case of `\enlargethispage` we will have infinite negative glue at the bottom of the page (coming from `\vss`) and that will earn us an error message if we `\vsplit` to get at the marks. So we need to remove the last glue (if any) at the end of `\@outputbox` as we are only interested in marks that change doesn't matter.

```

2096 \setbox\@outputbox\vbox{\unvbox\@outputbox\unskip}%
2097 \setbox\@outputbox\vsplit\@outputbox to\maxdimen

```

One minor difference from the current `fixmarks` package, pass the marks through a token register to stop any `#` tokens causing an error in a `\def`.

```

2098 \toks@\expandafter{\topmark}%
2099 \xdef\@firstcoltopmark{\the\toks@}%
2100 \toks@\expandafter{\splitfirstmark}%
2101 \xdef\@firstcolfirstmark{\the\toks@}%

```

This test does not work if truly empty marks have been inserted, but  $\LaTeX$  marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```

2102 \ifx\@firstcolfirstmark\@empty
2103 \global\let\@setmarks\relax
2104 \else
2105 \gdef\@setmarks{%
2106 \let\firstmark\@firstcolfirstmark

```

```

2107 \let\topmark\@firstcoltopmark}%
2108 \fi
 End of change
2109 \else
2110 \global\@firstcolumntrue
2111 \setbox\@outputbox\vbox{%
2112 \hb@xt@\textwidth{%
2113 \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
2114 \hfil
 The color of the \vrule should be \normalcolor as to not inherit the color from
 the column.
2115 {\normalcolor\vrule \@width\columnseprule}%
2116 \hfil
2117 \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
2118 (fltrace) \fl@trace{PAGE: second column also boxed}%
2119 \@combinedblfloats
 Override current first and top with those of first column if necessary
2120 \setmarks
 End of change
2121 \@outputpage
2122 (fltrace) \fl@trace{PAGE: two column page completed}%
2123 \begingroup
2124 \@dblfloatplacement
2125 \@startdblcolumn
2126 \@whiles\if@fcolmade \fi{\@outputpage
2127 (fltrace) \fl@trace{PAGE: double float page completed}%
2128 \@startdblcolumn}%
2129 \endgroup
2130 \fi}%
2131 (latexrelease | fltrace)\EndIncludeInRelease
2132 (latexrelease | fltrace)\IncludeInRelease{0000/00/00}%
2133 (latexrelease | fltrace) {\@outputdblcol}{2 column marks}%
2134 (latexrelease | fltrace)\def\@outputdblcol{%
2135 (latexrelease | fltrace) \if@firstcolumn
2136 (latexrelease | fltrace) \global \@firstcolumnfalse
2137 (latexrelease | fltrace) \global \setbox\@leftcolumn \box\@outputbox
2138 (*trace)
2139 (latexrelease | fltrace) \fl@trace{PAGE: first column boxed}%
2140 </trace>
2141 (latexrelease | fltrace) \else
2142 (latexrelease | fltrace) \global \@firstcolumntrue
2143 (latexrelease | fltrace) \setbox\@outputbox \vbox {%
2144 (latexrelease | fltrace) \hb@xt@\textwidth {%
2145 (latexrelease | fltrace) \hb@xt@\columnwidth {%
2146 (latexrelease | fltrace) \box\@leftcolumn \hss}%
2147 (latexrelease | fltrace) \hfil
2148 (latexrelease | fltrace) {\normalcolor\vrule
2149 (latexrelease | fltrace) \@width\columnseprule}%
2150 (latexrelease | fltrace) \hfil
2151 (latexrelease | fltrace) \hb@xt@\columnwidth {%
2152 (latexrelease | fltrace) \box\@outputbox \hss}%

```

```

2153 <latexrelease | fltrace> }%
2154 <latexrelease | fltrace> }%
2155 <*trace>
2156 <latexrelease | fltrace> \fl@trace{PAGE: second column also boxed}%
2157 </trace>
2158 <latexrelease | fltrace> \@combinedblfloats
2159 <latexrelease | fltrace> \@outputpage
2160 <*trace>
2161 <latexrelease | fltrace> \fl@trace{PAGE: two column page completed}%
2162 </trace>
2163 <latexrelease | fltrace> \begingroup
2164 <latexrelease | fltrace> \@dblfloatplacement
2165 <latexrelease | fltrace> \@startdblcolumn

```

This loop could be replaced by an `\expandafter` tail recursion in `\@startdblcolumn`.

```

2166 <latexrelease | fltrace> \@whilesw@if@fcolmade \fi
2167 <latexrelease | fltrace> {\@outputpage
2168 <*trace>
2169 <latexrelease | fltrace> \fl@trace{PAGE: double float page completed}%
2170 </trace>
2171 <latexrelease | fltrace> \@startdblcolumn}%
2172 <latexrelease | fltrace> \endgroup
2173 <latexrelease | fltrace> \fi
2174 <latexrelease | fltrace>}%
2175 <latexrelease | fltrace>\EndIncludeInRelease
2176 </2ekernel | fltrace | latexrelease>

```

### 64.1.3 Float placement parameters

The main purpose of this section is to ensure that all the float-placement parameters which need to be set in a class file or package have been declared. It also describes their use and sets values for them which are reasonable for typical documents using US letter or A4 sized paper.

#### Limits for the placement of floating objects

- `\c@topnumber` This counter holds the maximum number of floats that can appear at the top of a text page or column.
- ```

2177 <*2ekernel>
2178 \newcount\c@topnumber
2179 \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.
- ```

2180 \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.
- ```

2181 \newcount\c@bottomnumber
2182 \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.
2183 `\newcommand\bottomfraction{.3}`

`\c@totalnumber` This counter holds the maximum number of floats that can appear on any text page or column.
2184 `\newcount\c@totalnumber`
2185 `\setcounter{totalnumber}{3}`

`\textfraction` This macro holds the minimum proportion (as a decimal number) of a text page or column that must be occupied by text.
2186 `\newcommand\textfraction{.2}`

`\floatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page or column that must be occupied by floating objects before a ‘float page’ is produced.
2187 `\newcommand\floatpagefraction{.5}`

`\c@dbltopnumber` This counter holds the maximum number of double-column floats that can appear on the top of a two-column text page.
2188 `\newcount\c@dbltopnumber`
2189 `\setcounter{dbltopnumber}{2}`

`\dbltopfraction` This macro holds the maximum proportion (as a decimal number) of a two-column text page that can be occupied by double-column floats at the top.
2190 `\newcommand\dbltopfraction{.7}`

`\dblfloatpagefraction` This macro holds the minimum proportion (as a decimal number) of a page that must be occupied by double-column floating objects before a ‘double-column float page’ is produced.
2191 `\newcommand\dblfloatpagefraction{.5}`

Floats on a text page

`\floatsep` When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters
`\textfloatsep` are used for both one-column mode and single-column floats in two-column mode.
`\intextsep` They are all rubber lengths.

`\floatsep` is the space between adjacent floats that are placed at the top or bottom of the text page or column.

`\textfloatsep` is the space between the main text and floats at the top or bottom of the page or column.

`\intextsep` is the space between in-text floats and the text.

2192 `\newskip\floatsep`

2193 `\newskip\textfloatsep`

2194 `\newskip\intextsep`

2195 `\setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}`

2196 `\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}`

2197 `\setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}`

`\dblfloatsep` `\dbltextfloatsep` When double-column floats (floating objects that span the whole `\textwidth`) are placed at the top of a text page in two-column mode, the separation between the float and the text is controlled by `\dblfloatsep` and `\dbltextfloatsep`. They are rubber lengths.

`\dblfloatsep` is the space between adjacent double-column floats placed at the top of the text page.

`\dbltextfloatsep` is the space between the main text and double-column floats at the top of the page.

```
2198 \newskip\dblfloatsep
2199 \newskip\dbltextfloatsep
2200 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
2201 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
```

Floats on their own page or column

`\@fptop` `\@fpsep` `\@fpbot` When floating objects are placed on a separate page or column, called a ‘float page’, the layout of the page is controlled by these parameters, which are rubber lengths.

At the top of the page `\@fptop` is inserted; typically this supplies some stretchable whitespace. At the bottom of the page `\@fpbot` is inserted. Between adjacent floats `\@fpsep` is inserted.

These parameters are used for all floating objects on a ‘float page’ in one-column mode, and for single-column floats in two-column mode.

Note that at least one of the two parameters `\@fptop` and `\@fpbot` should contain a `plus ...fil` so as to fill the remaining empty space.

```
2202 \newskip\@fptop
2203 \newskip\@fpsep
2204 \newskip\@fpbot
2205 \setlength\@fptop{0\p@ \@plus 1fil}
2206 \setlength\@fpsep{8\p@ \@plus 2fil}
2207 \setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblfpptop` Double-column ‘float pages’ in two-column mode use similar parameters.

```
\@dblfpsep 2208 \newskip\@dblfpptop
\@dblfpbot 2209 \newskip\@dblfpsep
2210 \newskip\@dblfpbot
2211 \setlength\@dblfpptop{0\p@ \@plus 1fil}
2212 \setlength\@dblfpsep{8\p@ \@plus 2fil}
2213 \setlength\@dblfpbot{0\p@ \@plus 1fil}
```

`\topfigrule` `\botfigrule` The macros can be used to put in rules between floats and text; whatever they insert should be vertical mode material which takes up zero space.

```
\dblfigrule 2214 \let\topfigrule=\relax
2215 \let\botfigrule=\relax
2216 \let\dblfigrule=\relax
2217 /</2kernel)
```

File L

ltxclass.dtx

65 Introduction

This file implements the following declarations, which replace `\documentstyle` in $\text{\LaTeX} 2_{\epsilon}$ documents.

Note that old documents containing `\documentstyle` will be run using a compatibility option—thus keeping everyone happy, we hope!

The overall idea is that there are two types of ‘style files’: ‘class files’ which define elements and provide a default formatting for them; and ‘packages’ which provide extra functionality. One difference between $\text{\LaTeX} 2_{\epsilon}$ and $\text{\LaTeX} 2.09$ is that $\text{\LaTeX} 2_{\epsilon}$ packages may have options. Note that options to classes packages may be implemented such that they input files, but these file names are not necessarily directly related to the option name.

66 User interface

```
\documentclass[<main-option-list>]{<class>}[<version>]
```

There must be exactly one such declaration, and it must come first. The *<main-option-list>* is a list of options which can modify the formatting of elements which are defined in the *<class>* file as well as in all following `\usepackage` declarations (see below). The *<version>* is a version number, beginning with a date in the format YYYY/MM/DD. If an older version of the class is found, a warning is issued.

```
\documentstyle[<main-option-list>]{<class>}[<version>]
```

The `\documentstyle` declaration is kept in order to maintain upward compatibility with $\text{\LaTeX} 2.09$ documents. It is similar to `\documentclass`, but it causes all options in *<main-option-list>* that the *<class>* does not use to be passed to `\RequirePackage` after the options have been processed. This maintains compatibility with the 2.09 behaviour. Also a flag is set to indicate that the document is to be processed in $\text{\LaTeX} 2.09$ compatibility mode. As far as most packages are concerned, this only affects the warnings and errors \LaTeX generates. This flag does affect the definition of font commands, and `\sloppy`.

```
\usepackage[<package-option-list>]{<package-list>}[<version>]
```

There can be any number of these declarations. All packages in *<package-list>* are called with the same options.

Each *<package>* file defines new elements (or modifies those defined in the *<class>*), and thus extends the range of documents which can be processed. The *<package-option-list>* is a list of options which can modify the formatting of elements defined in the *<package>* file. The *<version>* is a version number, beginning with a date in the format YYYY/MM/DD. If an older version of the package is found, a warning is issued.

Each package is loaded only once. If the same package is requested more than once, nothing happens, unless the package has been requested with options that were not given the first time it was loaded, in which case an error is produced.

As well as processing the options given in the $\langle package-option-list \rangle$, each package processes the $\langle main-option-list \rangle$. This means that options that affect all of the packages can be given globally, rather than repeated for every package.

Note that class files have the extension `.cls`, packages have the extension `.sty`.

`filecontents`

The environment `filecontents` is intended for passing the contents of packages, options, or other files along with a document in a single file. It has one argument, which is the name of the file to create. If that file already exists (maybe only in the current directory if the OS supports a notion of a ‘current directory’ or ‘default directory’) then nothing happens (except for an information message) and the body of the environment is bypassed. Otherwise, the body of the environment is written verbatim to the file name given as the first argument, together with some comments about how it was produced.

The environment is allowed only before `\documentclass` to ensure that all packages or options necessary for this particular run are present when needed. The begin and end tags should each be on a line by itself. There is also a star-form; this does not write extra comments into the file.

66.1 Option processing

When the options are processed, they are divided into two types: *local* and *global*:

- For a class, the options in the `\documentclass` command are local.
- For a package, the options in the `\usepackage` command are local, and the options in the `\documentclass` command are global.

The options for `\documentclass` and `\usepackage` are processed in the following way:

1. The local and global options that have been declared (using `\DeclareOption` as described below) are processed first.

In the case of `\ProcessOptions`, they are processed in the order that they were declared in the class or package.

In the case of `\ProcessOptions*`, they are processed in the order that they appear in the option-lists. First the global options, and then the local ones.

2. Any remaining local options are dealt with using the default option (declared using the `\DeclareOption*` declaration described below). For document classes, this usually does nothing, but records the option on a list of unused options. For packages, this usually produces an error.

Finally, when `\begin{document}` is reached, if there are any global options which have not been used by either the class or any package, the system will produce a warning.

67 Class and Package interface

67.1 Class name and version

`\ProvidesClass`

A class can identify itself with the `\ProvidesClass{\langle name \rangle}[\langle version \rangle]` command. The $\langle version \rangle$ should begin with a date in the format YYYY/MM/DD.

67.2 Package name and version

`\ProvidesPackage` A package can identify itself with the `\ProvidesPackage{<name>}[<version>]` command. The `<version>` should begin with a date in the format YYYY/MM/DD.

67.3 Requiring other packages

`\RequirePackage` Packages or classes can load other packages using `\RequirePackage[<options>]{<name>}[<version>]`. If the package has already been loaded, then nothing happens unless the requested options are not a subset of the options with which it was loaded, in which case an error is called.

`\LoadClass` Similar to `\RequirePackage`, but for classes, may not be used in package files.

`\PassOptionsToPackage` Packages can pass options to other packages using:

`\PassOptionsToPackage{<options>}{<package>}`.

`\PassOptionsToClass` This adds the `<options>` to the options list of any future `\RequirePackage` or `\usepackage` command. For example:

```
\PassOptionsToPackage{foo,bar}{fred}
\RequirePackage[baz]{fred}
```

is the same as:

```
\RequirePackage[foo,bar,baz]{fred}
```

`\LoadClassWithOptions` `\LoadClassWithOptions{<name>}[<version>]:`

This is similar to `\LoadClass`, but it always calls class `<name>` with exactly the same option list that is being used by the current class, rather than an option explicitly supplied or passed on by `\PassOptionsToClass`.

`\RequirePackageWithOptions` `\RequirePackageWithOptions` is the analogous command for packages.

This is mainly intended to allow one class to simply build on another, for example:

```
\LoadClassWithOptions{article}
```

This should be contrasted with the slightly different construction

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
\LoadClass{article}
```

As used here, the effects are more or less the same, but the version using `\LoadClassWithOptions` is slightly quicker (and less to type). If, however, the class declares options of its own then the two constructions are different; compare, for example:

```
\DeclareOption{landscape}{...}
\ProcessOptions
\LoadClassWithOptions{article}
```

with:

```
\DeclareOption{landscape}{...}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions
\LoadClass{article}
```


In the first case, the `article` class will be called with option `landscape` precisely when the current class is called with this option; but in the second example it will not as in that case `article` is only passed options by the default option handler, which is not used for `landscape` as that option is explicitly declared.

`\@ifpackageloaded` To find out if a package has already been loaded, use
`\@ifclassloaded` `\@ifpackageloaded{<package>}{<true>}{<false>}`.
`\@ifpackagelater` To find out if a package has already been loaded with a version equal to or more recent than `<version>`, use
`\@ifclasslater` `\@ifpackagelater{<package>}{<version>}{<true>}{<false>}`.
`\@ifpackagewith` To find out if a package has already been loaded with at least the options
`\@ifclasswith` `<options>`, use `\@ifpackagewith{<package>}{<options>}{<true>}{<false>}`.

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).

67.4 Declaring new options

Options for classes and packages are built using the same macros.

`\DeclareOption` To define a builtin option, use `\DeclareOption{<name>}{<code>}`.
`\DeclareOption*` To define the default action to perform for local options which have not been declared, use `\DeclareOption*{<code>}`.

Note: there should be no use of

`\RequirePackage`, `\DeclareOption`, `\DeclareOption*` or `\ProcessOptions` inside `\DeclareOption` or `\DeclareOption*`.

Possible uses for `\DeclareOption*` include:

`\DeclareOption*{}`

Do nothing. Silently accept unknown options. (This suppresses the usual warnings.)

`\DeclareOption*{\@unknownoptionerror}`

Complain about unknown local options. (The initial setting for package files.)

`\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{<pkg-name>}`

Handle the the current option by passing it on to the package `<pkg-name>`, which will presumably be loaded via `\RequirePackage` later in the file. This is useful for building ‘extension’ packages, that perhaps handle a couple of new options, but then pass everything else on to an existing package.

`\DeclareOption*{\InputIfFileExists{xx-CurrentOption.yyy}%
 {}%
 {\OptionNotUsed}}`

Handle the option `foo` by loading the file `xx-foo.yyy` if it exists, otherwise do nothing, but declare that the option was not used. Actually the `\OptionNotUsed` declaration is only needed if this is being used in class files, but does no harm in package files.

67.5 Safe Input Macros

`\InputIfFileExists` `\InputIfFileExists{<file>}{<then>}{<else>}`
 Inputs `<file>` if it exists. Immediately before the input, `<then>` is executed. Otherwise `<else>` is executed.
`\IfFileExists` As above, but does not input the file.
 One thing you might like to put in the `<else>` clause is

`\@missingfileerror` This starts an interactive request for a filename, supplying default extensions. Just hitting return causes the whole input to be skipped and entering x quits the current run,

`\input` This has been redefined from the L^AT_EX2.09 definition, in terms of the new commands `\InputIfFileExists` and `\@missingfileerror`.

`\listfiles` 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 `\ProvidesPackage` are listed alongside the file name. So files in standard (and other non-standard) distributions can put informative strings in this argument.

68 Implementation

```

1 (*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 \@currentt.
\@pkgextension 16 \def\@clsextension{cls}
                17 \def\@pkgextension{sty}
                18 \@onlypreamble\@clsextension
                19 \@onlypreamble\@pkgextension

\@pushfilename Commands to push and pop the file name and extension.
\@popfilename  #1 current name.
\@currnamestack #2 current extension.
                #3 current catcode of @.
                #4 Rest of the stack.
                20 \def\@pushfilename{%
                21   \xdef\@currnamestack{%
                22     {\@currname}%
                23     {\@currentt}%
                24     {\the\catcode'\@}%
                25     \@currnamestack}}
                26 \@onlypreamble\@pushfilename
                27 \def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil}
                28 \@onlypreamble\@popfilename
                29 \def\@p@pfilename#1#2#3#4\@nil{%
                30   \gdef\@currname{#1}%
                31   \gdef\@currentt{#2}%
                32   \catcode'\@#3\relax
                33   \gdef\@currnamestack{#4}}
                34 \@onlypreamble\@p@pfilename
                35 \gdef\@currnamestack{}
                36 \@onlypreamble\@currnamestack

\@optionlist Returns the option list of the file.
                37 \def\@optionlist#1{%
                38   \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}}
                39 \@onlypreamble\@optionlist

\@ifpackageloaded \@ifpackageloaded{<name>} Checks to see whether a file has been loaded.
\@ifclassloaded  40 \def\@ifpackageloaded{\@ifl@aded\@pkgextension}
                  41 \def\@ifclassloaded{\@ifl@aded\@clsextension}
                  42 \@onlypreamble\@ifpackageloaded
                  43 \@onlypreamble\@ifclassloaded
                  44 \def\@ifl@aded#1#2{%
                  45   \expandafter\ifx\csname ver@#2.#1\endcsname\relax
                  46   \expandafter\@secondoftwo
                  47   \else
                  48   \expandafter\@firstoftwo
                  49   \fi}
                  50 \@onlypreamble\@ifl@aded

\@ifpackagelater \@ifpackagelater{<name>}{YYYY/MM/DD} Checks that the package loaded is
\@ifclasslater   more recent than the given date.
                  51 \def\@ifpackagelater{\@ifl@ter\@pkgextension}
                  52 \def\@ifclasslater{\@ifl@ter\@clsextension}
                  53 \@onlypreamble\@ifpackagelater
                  54 \@onlypreamble\@ifclasslater

```

```

55 \def\@ifl@ter#1#2{%
56   \expandafter\@ifl@t@r
57   \csname ver@#2.#1\endcsname}
58 \@onlypreamble\@ifl@ter

```

This internal macro is also used in `\NeedsTeXFormat`.

```

59 \def\@ifl@t@r#1#2{%
60   \ifnum\expandafter\@parse@version#1//00\@nil<%
61     \expandafter\@parse@version#2//00\@nil
62   \expandafter\@secondoftwo
63   \else
64     \expandafter\@firstoftwo
65   \fi}
66 \@onlypreamble\@ifl@t@r

67 \def\@parse@version#1/#2/#3#4#5\@nil{#1#2#3#4 }
68 \@onlypreamble\@parse@version

```

`\@ifpackagewith` `\@ifclasswith` `\@ifpackagewith{<name>}{<option-list>}` Checks that `<option-list>` is a subset of the options `with` which `<name>` was loaded.

```

69 \def\@ifpackagewith{\@ifoptions\@pkgextension}
70 \def\@ifclasswith{\@ifoptions\@clsextension}
71 \@onlypreamble\@ifpackagewith
72 \@onlypreamble\@ifclasswith

73 \def\@ifoptions#1#2{%
74   \@expandtwoargs\@if@ptions{\@optionlist{#2.#1}}
75 \@onlypreamble\@ifoptions

```

Probably shouldn't use `\CurrentOption` here... (changed to `\reserved@b`.)

```

76 \def\@if@ptions#1#2{%
77   \let\reserved@a\@firstoftwo
78   \@for\reserved@b:=#2\do{%
79     \ifx\reserved@b\@empty
80     \else
81       \expandafter\in@\expandafter{\expandafter,\reserved@b,}{, #1,}%
82       \ifin@
83       \else
84         \let\reserved@a\@secondoftwo
85       \fi
86     \fi
87   }%
88 \reserved@a}
89 \@onlypreamble\@if@ptions

```

`\ProvidesPackage` Checks that the current filename is correct, and defines `\ver@filename`.

```

90 \def\ProvidesPackage#1{%
91   \xdef\@gtempa{#1}%
92   \ifx\@gtempa\@currname\else
93     \@latex@warning@no@line{You have requested
94       \@cls@pkg\space'\@currname',\MessageBreak
95       but the \@cls@pkg\space provides '#1'}%
96   \fi
97   \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}]{}%
98 \@onlypreamble\ProvidesPackage

```

```

99 \def\@pr@videpackage[#1]{%
100 \expandafter\xdef\csname ver@\currname.\@current\endcsname{#1}%
101 \ifx\@current\@clsextension
102 \typeout{Document Class: \@gtempa\space#1}%
103 \else
104 \wlog{Package: \@gtempa\space#1}%
105 \fi}
106 \@onlypreamble\@pr@videpackage

```

`\ProvidesClass` Like `\ProvidesPackage`, but for classes.

```

107 \let\ProvidesClass\ProvidesPackage
108 \@onlypreamble\ProvidesClass

```

`\ProvidesFile` Like `\ProvidesPackage`, but for arbitrary files. Do not apply `\@onlypreamble` to these, as we may want to label files input during the document.

`\@providesfile`

```

109 \def\ProvidesFile#1{%
110 \begingroup
111 \catcode'\ 10 %
112 \ifnum \endlinechar<256 %
113 \ifnum \endlinechar>\m@ne
114 \catcode\endlinechar 10 %
115 \fi
116 \fi
117 \@makeother\/%
118 \@makeother\&%
119 \kernel@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}]

```

During `intex` a special version of `\@providesfile` is used. The real definition is installed right at the end, in `ltfinal.dtx`.

```

\def\@providesfile#1[#2]{%
\wlog{File: #1 #2}%
\expandafter\xdef\csname ver@#1\endcsname{#2}%
\endgroup}
\end{macrocode}

```

`\PassOptionsToPackage` If the package has been loaded, we check that it was first loaded with the options.

`\PassOptionsToClass` Otherwise we add the option list to that of the package.

```

120 \def\@pass@options#1#2#3{%
121 \expandafter\xdef\csname opt@#3.#1\endcsname{%
122 \@ifundefined{opt@#3.#1}\@empty
123 {\csname opt@#3.#1\endcsname,}%
124 \zap@space#2 \@empty}}
125 \@onlypreamble\@pass@options
126 \def\PassOptionsToPackage{\@pass@options\@pkgextension}
127 \def\PassOptionsToClass{\@pass@options\@clsextension}
128 \@onlypreamble\PassOptionsToPackage
129 \@onlypreamble\PassOptionsToClass

```

`\DeclareOption` Adds an option as a `\ds@` command, or the default `\default@ds` command.

`\DeclareOption*` 130 `\def\DeclareOption{%`

```

131 \let\@fileswith@pti@ns\@badrequireerror
132 \@ifstar\@defdefault@ds\@declareoption}
133 \long\def\@declareoption#1#2{%
134   \xdef\@declaredoptions{\@declaredoptions,#1}%
135   \toks@{#2}%
136   \expandafter\edef\csname ds@#1\endcsname{\the\toks@}}
137 \long\def\@defdefault@ds#1{%
138   \toks@{#1}%
139   \edef\default@ds{\the\toks@}}
140 \@onlypreamble\DeclareOption
141 \@onlypreamble\@declareoption
142 \@onlypreamble\@defdefault@ds

```

`\OptionNotUsed` If we are in a class file, add `\CurrentOption` to the list of unused options. Otherwise, in a package file do nothing.

```

143 \def\OptionNotUsed{%
144   \ifx\@current\@clsextension
145     \xdef\@unusedoptionlist{%
146       \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
147       \CurrentOption}%
148   \fi}
149 \@onlypreamble\OptionNotUsed

```

`\default@ds` The default default option code. Set by `\@onefilewithoptions` to either `\OptionNotUsed` for classes, or `\@unknownoptionerror` for packages. This may be reset in either case with `\DeclareOption*`.

```
150 % \let\default@ds\OptionNotUsed
```

`\ProcessOptions` `\ProcessOptions` calls `\ds@option` for each known package option, then calls `\default@ds` for each option on the local options list. Finally resets all the declared options to `\relax`. The empty option does nothing, this has to be reset on the off chance it's set to `\relax` if an empty element gets into the `\@declaredoptions` list.

The star form is similar but executes options given in the order specified in the document, not the order they are declared in the file. In the case of packages, global options are executed before local ones.

```

151 \def\ProcessOptions{%
152   \let\ds@\@empty
153   \edef\@curroptions{\@ptionlist{\@currname.\@current}}%
154   \@ifstar\@xprocessoptions\@processoptions}
155 \@onlypreamble\ProcessOptions

156 \def\@processoptions{%
157   \@for\CurrentOption:=\@declaredoptions\do{%
158     \ifx\CurrentOption\@empty\else
159       \@expandtwoargs\in@{,\CurrentOption,}%
160       ,\ifx\@current\@clsextension\else\@classoptionslist,\fi
161       \@curroptions,}%
162   \ifin@
163     \@useoption
164     \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
165   \fi
166   \fi}%

```

```

167 \@process@pti@ns}
168 \@onlypreamble\@process@ptions

169 \def\@xprocess@ptions{%
170 \ifx\@current\@clsextension\else
171 \for\CurrentOption:=\@classoptionslist\do{%
172 \ifx\CurrentOption\@empty\else
173 \@expandtwoargs\in@{\, \CurrentOption,}{\, \@declaredoptions,}%
174 \ifin@
175 \use@option
176 \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
177 \fi
178 \fi}%
179 \fi
180 \@process@pti@ns}
181 \@onlypreamble\@xprocess@ptions

```

The common part of `\ProcessOptions` and `\ProcessOptions*`.

```

182 \def\@process@pti@ns{%
183 \for\CurrentOption:=\@curroptions\do{%
184 \@ifundefined{ds@\CurrentOption}%
185 {\use@option
186 \default@ds}%

```

There should not be any non-empty definition of `\CurrentOption` at this point, as all the declared options were executed earlier. This is for compatibility with 2.09 styles which use `\def\ds@...` directly, and so have options which do not appear in `\@declaredoptions`.

```

187 \use@option}%

```

Clear all the definitions for option code. First set all the declared options to `\relax`, then reset the ‘default’ and ‘empty’ options. and the list of declared options.

```

188 \for\CurrentOption:=\@declaredoptions\do{%
189 \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%

190 \let\CurrentOption\@empty
191 \let\@fileswith@pti@ns\@fileswith@pti@ns
192 \AtEndOfPackage{\let\@unprocessedoptions\relax}}
193 \@onlypreamble\@process@pti@ns

```

`\@options` `\@options` is a synonym for `\ProcessOptions*` for upward compatibility with L^AT_EX 2.09 style files.

```

194 \def\@options{\ProcessOptions*}
195 \@onlypreamble\@options

```

`\use@option` Execute the code for the current option.

```

196 \def\use@option{%
197 \@expandtwoargs\@removeelement\CurrentOption
198 \@unusedoptionlist\@unusedoptionlist
199 \csname ds@\CurrentOption\endcsname}
200 \@onlypreamble\use@option

```

`\ExecuteOptions` `\ExecuteOptions{<option-list>}` executes the code declared for each option.

```

201 \def\ExecuteOptions#1{%

```

```

202 \def\reserved@a##1\@nil{%
203   \for\CurrentOption:=#1\do{\csname ds@\CurrentOption\endcsname}%
204   \edef\CurrentOption{##1}}%
205 \expandafter\reserved@a\CurrentOption\@nil}
206 \@onlypreamble\ExecuteOptions

```

The top-level commands, which just set some parameters then call the internal command, \@fileswithoptions.

`\documentclass` The main new-style class declaration.

```

207 \def\documentclass{%
208   \let\documentclass\@twoclasseserror
209   \if@compatibility\else\let\usepackage\RequirePackage\fi
210   \@fileswithoptions\@clsextension}
211 \@onlypreamble\documentclass

```

`\documentstyle` 2.09 style class ‘style’ declaration.

```

212 \def\documentstyle{%
213   \makeatletter\input{latex209.def}\makeatother
214   \documentclass}
215 \@onlypreamble\documentstyle

```

`\RequirePackage` Load package if not already loaded.

```

216 \def\RequirePackage{%
217   \@fileswithoptions\@pkgextension}
218 \@onlypreamble\RequirePackage

```

`\LoadClass` Load class.

```

219 \def\LoadClass{%
220   \ifx\@current\@pkgextension
221     \@latex@error
222     {\noexpand\LoadClass in package file}%
223     {You may only use \noexpand\LoadClass in a class file.}%
224   \fi
225   \@fileswithoptions\@clsextension}
226 \@onlypreamble\LoadClass

```

`\@loadwithoptions` Pass the current option list on to a class or package. #1 is \@cls-or-pkgextension, #2 is \RequirePackage or \LoadClass, #3 is the class or package to be loaded.

```

227 \def\@loadwithoptions#1#2#3{%
228   \expandafter\let\csname opt@#3.#1\expandafter\endcsname
229   \csname opt@\@currname.\@current\endcsname
230   #2{#3}}
231 \@onlypreamble\@loadwithoptions

```

`\LoadClassWithOptions` Load class ‘#1’ with the current option list.

```

232 \def\LoadClassWithOptions{%
233   \@loadwithoptions\@clsextension\LoadClass}
234 \@onlypreamble\LoadClassWithOptions

```

`\RequirePackageWithOptions` Load package ‘#1’ with the current option list.

```

235 \def\RequirePackageWithOptions{%
236   \AtEndOfPackage{\let\@unprocessedoptions\relax}%
237   \@loadwithoptions\@pkgextension\RequirePackage}
238 \@onlypreamble\RequirePackageWithOptions

```


`\usepackage` To begin with, `\usepackage` produces an error. This is reset by `\documentclass`.

```
239 \def\usepackage#1{%
240   \latex@error
241     {\noexpand \usepackage before \string\documentclass}%
242     {\noexpand \usepackage may only appear in the document
243      preamble, i.e.,\MessageBreak
244      between \noexpand\documentclass and
245      \string\begin{document}.}%
246   \@gobble}
247 \@onlypreamble\usepackage
```

`\NeedsTeXFormat` Check that the document is running on the correct system.

```
248 \def\NeedsTeXFormat#1{%
249   \def\reserved@a{#1}%
250   \ifx\reserved@a\fmtname
251     \expandafter\@needsformat
252   \else
253     \latex@error{This file needs format ‘\reserved@a’%
254       \MessageBreak but this is ‘\fmtname’}%
255     The current input file will not be processed
256     further,\MessageBreak
257     because it was written for some other flavor of
258     TeX.\MessageBreak\@ehd}%
```

If the file is not meant to be processed by L^AT_EX 2_ε we stop inputting it, but we do not end the run. We just end inputting the current file.

```
259   \endinput \fi}
260 \@onlypreamble\NeedsTeXFormat

261 \def\@needsformat{%
262   \@ifnextchar[%
263     \@needsformat
264   {}]
265 \@onlypreamble\@needsformat

266 \def\@needsformat[#1]{%
267   \@ifl@t@r\fmtversion{#1}{}%
268   {\latex@warning@no@line
269     {You have requested release ‘#1’ of LaTeX,\MessageBreak
270     but only release ‘\fmtversion’ is available}}
271 \@onlypreamble\@needsformat
```

`\zap@space` `\zap@space foo<space>\@empty` removes all spaces from `foo` that are not protected by `{ }` groups.

```
272 \def\zap@space#1 #2{%
273   #1%
274   \ifx#2\@empty\else\expandafter\zap@space\fi
275   #2}
```

`\@fileswithoptions` The common part of `\documentclass` and `\usepackage`.

```
276 \def\@fileswithoptions#1{%
277   \@ifnextchar[%
278     {\@fileswithoptions#1}%
```

```

279     {\@fileswith@ptions#1[]}}
280 \@onlypreamble\@fileswithoptions

281 \def\@fileswith@ptions#1[#2]#3{%
282   \ifnextchar[%]
283     {\@fileswith@ptions#1[#{2}]#3}%
284     {\@fileswith@ptions#1[#{2}]#3[]}}
285 \@onlypreamble\@fileswithoptions

```

Then we do some work.

First of all, we define the global variables. Then we look to see if the file has already been loaded. If it has, we check that it was first loaded with at least the current options. If it has not, we add the current options to the package options, set the default version to be 0000/00/00, and load the file if we can find it. Then we check the version number.

Finally, we restore the old file name, reset the default option, and we set the catcode of @.

For classes, we can immediately process the file. For other types, #2 could be a comma separated list, so loop through, processing each one separately.

```

286 \def\@fileswith@ptions#1[#2]#3[#4]{%
287   \ifx#1\@clsextension
288     \ifx\@classoptionslist\relax
289       \xdef\@classoptionslist{\zap@space#2 \@empty}%
290       \def\reserved@a{%
291         \@onefilewithoptions#3[#{2}][#{4}]#1%
292         \@documentclasshook}%
293       \else
294         \def\reserved@a{%
295           \@onefilewithoptions#3[#{2}][#{4}]#1%
296         \fi
297       \else

```

build up a list of calls to \@onefilewithoptions (one for each package) without thrashing the parameter stack.

```

298   \def\reserved@b##1,{%
299     \ifx\@nil##1\relax\else
300       \ifx\relax##1\relax\else
301         \noexpand\@onefilewithoptions##1[#{2}][#{4}]%
302         \noexpand\@pkgextension
303       \fi
304       \expandafter\reserved@b
305     \fi}%
306   \edef\reserved@a{\zap@space#3 \@empty}%
307   \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nil,}%
308   \fi
309   \reserved@a}
310 \@onlypreamble\@fileswithoptions

```

Have the main argument as #1, so we only need one \@expandafter above.

```

311 \def\@onefilewithoptions#1[#2][#3]#4{%
312   \@pushfilename
313   \xdef\@currname{#1}%
314   \global\let\@current#4%
315   \expandafter\let\csname\@currname.\@current-h@k\endcsname\@empty
316   \let\CurrentOption\@empty

```

```

317 \@reset@options
318 \makeatletter

Grab everything in a macro, so the parameter stack is popped before any processing begins.

319 \def\reserved@a{%
320   \ifl@aded\@currentx{#1}%
321   {\@if@options\@currentx{#1}{#2}{-}}%
322   {\@latex@error
323     {Option clash for \@cls@pkg\space #1}%
324     {The package #1 has already been loaded
325       with options:\MessageBreak
326       \space\space[\@optionlist{#1.\@currentx}]\MessageBreak
327       There has now been an attempt to load it
328         with options\MessageBreak
329         \space\space[#2]\MessageBreak
330         Adding the global options:\MessageBreak
331         \space\space
332         \@optionlist{#1.\@currentx},#2\MessageBreak
333         to your \noexpand\documentclass declaration may fix this.%
334         \MessageBreak
335         Try typing \space <return> \space to proceed.}}}%
336   {\@pass@options\@currentx{#2}{#1}%

337   \global\expandafter
338   \let\csname ver@\@currname.\@currentx\endcsname\@empty
339   \InputIfFileExists
340     {\@currname.\@currentx}%
341     {}%
342     {\@missingfileerror\@currname\@currentx}%

\@unprocessedoptions will generate an error for each specified option in a package unless a \ProcessOptions has appeared in the package file.

343   \let\@unprocessedoptions\@unprocessedoptions
344   \csname\@currname.\@currentx-h@k\endcsname
345   \expandafter\let\csname\@currname.\@currentx-h@k\endcsname
346     \undefined
347   \@unprocessedoptions}

348   \@ifl@ter\@currentx{#1}{#3}{-}}%
349   {\@latex@warning@no@line
350     {You have requested,\on@line,
351     version\MessageBreak
352     ‘#3’ of \@cls@pkg\space #1,\MessageBreak
353     but only version\MessageBreak
354     ‘\csname ver@#1.\@currentx\endcsname’\MessageBreak
355     is available}}}%

356   \ifx\@currentx\@clsextension\let\LoadClass\@twoloadclasserror\fi
357   \@popfilename
358   \@reset@options}%
359 \reserved@a}
360 \@onlypreamble\@onefilewithoptions

```

`\@fileswith@ptions` Save the definition (for error checking).

```

361 \let@@fileswith@pti@ns\@fileswith@pti@ns
362 \@onlypreamble\@fileswith@pti@ns

```

`\@reset@ptions` Reset the default option, and clear lists of declared options.

```

363 \def\@reset@ptions{%
364   \global\ifx\@current\@clsextension
365     \let\default@ds\OptionNotUsed
366   \else
367     \let\default@ds\@unknownoptionerror
368   \fi
369   \global\let\ds@\@empty
370   \global\let\@declaredoptions\@empty}
371 \@onlypreamble\@reset@ptions

```

68.1 Hooks

Allow code do be saved to be executed at specific later times.

Save things in macros, I considered using toks registers, (and `\addto@hook` from the NFSS code, that would require stacking the contents in the case of required packages, so just generate a new macro for each package.

`\@begindocumenthook` Stuff to appear at the beginning or end of the document.

```

\@enddocumenthook 372 \ifx\@begindocumenthook\@undefined
373   \let\@begindocumenthook\@empty
374 \fi
375 \let\@enddocumenthook\@empty

```

`\g@addto@macro` Globally add to the end of a macro.

```

376 \long\def\g@addto@macro#1#2{%
377   \begingroup
378   \toks@\expandafter{#1#2}%
379   \xdef#1{\the\toks@}%
380   \endgroup}

```

`\AtEndOfPackage` The access functions.

```

\AtEndOfClass 381 \def\AtEndOfPackage{%
\AtBeginDocument 382   \expandafter\g@addto@macro\csname\@currname.\@current-h@k\endcsname}
\AtEndDocument 383 \let\AtEndOfClass\AtEndOfPackage
384 \@onlypreamble\AtEndOfPackage
385 \@onlypreamble\AtEndOfClass

386 \def\AtBeginDocument{\g@addto@macro\@begindocumenthook}
387 \def\AtEndDocument{\g@addto@macro\@enddocumenthook}
388 \@onlypreamble\AtBeginDocument

```

`\@cls@pkg` The current file type.

```

389 \def\@cls@pkg{%
390   \ifx\@current\@clsextension
391     document class%
392   \else
393     package%
394   \fi}
395 \@onlypreamble\@cls@pkg

```

```

\@unknownoptionerror Bad option.
396 \def\@unknownoptionerror{%
397   \latex@error
398   {Unknown option ‘\CurrentOption’ for \cls@pkg\space‘\currname’}%
399   {The option ‘\CurrentOption’ was not declared in
400     \cls@pkg\space‘\currname’, perhaps you\MessageBreak
401     misspelled its name.
402     Try typing \space <return>
403     \space to proceed.}}
404 \onlypreamble\@unknownoptionerror

\@unprocessedoptions Declare an error for each option, unless a \ProcessOptions occurred.
405 \def\@unprocessedoptions{%
406   \ifx\@current\@pkgextension
407     \edef\@curroptions{\@optionlist{\currname.\current}}%
408     \for\CurrentOption:=\@curroptions\do{%
409       \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
410   \fi}
411 \onlypreamble\@unprocessedoptions
412 \onlypreamble\@unprocessedoptions

\@badrequireerror \RequirePackage or \LoadClass occurs in the options section.
413 \def\@badrequireerror#1[#2]#3[#4]{%
414   \latex@error
415   {\noexpand\RequirePackage or \noexpand\LoadClass
416     in Options Section}%
417   {The \cls@pkg\space ‘\currname’ is defective.\MessageBreak
418     It attempts to load ‘#3’ in the options section, i.e.,\MessageBreak
419     between \noexpand\DeclareOption and \string\ProcessOptions.}}
420 \onlypreamble\@badrequireerror

\@twoloadclasserror Two \LoadClass in a class.
421 \def\@twoloadclasserror{%
422   \latex@error
423   {Two \noexpand\LoadClass commands}%
424   {You may only use one \noexpand\LoadClass in a class file}}
425 \onlypreamble\@twoloadclasserror

\@twoclasseserror Two \documentclass or \documentstyle.
426 \def\@twoclasseserror#1#{%
427   \latex@error
428   {Two \noexpand\documentclass or \noexpand\documentstyle commands}%
429   {The document may only declare one class.}\@gobble}
430 \onlypreamble\@twoclasseserror

```

68.2 Providing shipment

```

\two@digits Prefix a number less than 10 with ‘0’.
431 \def\two@digits#1{\ifnum#1<10 0\fi\number#1}

\filecontents This environment implements inline files. The star-form does not write extra
\endfilecontents comments into the file.

```

```

432 \begingroup%
433 \catcode'\*=11 %
434 \catcode'\^^M\active%
435 \catcode'\^^L\active\let^^L\relax%
436 \catcode'\^^I\active%

437 \gdef\filecontents{\@tempwatrue\filec@ntents}%
438 \gdef\filecontents*{\@tempwafalse\filec@ntents}%

439 \gdef\filec@ntents#1{%
440   \openin\@inputcheck#1 %
441   \ifeof\@inputcheck%
442     \@latex@warning@no@line%
443       {Writing file '\@currdir#1'}%

444   \chardef\reserved@c15 %
445   \ch@ck7\reserved@c\write%
446   \immediate\openout\reserved@c#1\relax%
447   \else%

448   \closein\@inputcheck%
449   \@latex@warning@no@line%
450     {File '#1' already exists on the system.\MessageBreak%
451     Not generating it from this source}%
452   \let\write\@gobbletwo%
453   \let\closeout\@gobble%
454   \fi%
455   \if@tempswa%

456   \immediate\write\reserved@c{%
457     \@percentchar\@percentchar\space%
458     \expandafter\@gobble\string\LaTeX2e file '#1'^^J%
459     \@percentchar\@percentchar\space generated by the %
460     '\@currenvir' \expandafter\@gobblefour\string\newenvironment^^J%
461     \@percentchar\@percentchar\space from source '\jobname' on %
462     \number\year/\two@digits\month/\two@digits\day.^^J%
463     \@percentchar\@percentchar}%
464   \fi%
465   \let\do\@makeother\dospecials%

466 \edef\E{\@backslashchar end\string{\@currenvir\string}}%
467 \edef\reserved@b{%
468   \def\noexpand\reserved@b%
469     #####1\E#####2\E#####3\relax}%
470 \reserved@b{%
471   \ifx\relax###3\relax%

There was no \end{filecontents}

472   \immediate\write\reserved@c{##1}%
473   \else%

There was a \end{filecontents}, so stop this time.

474   \edef^^M{\noexpand\end{\@currenvir}}%
475   \ifx\relax##1\relax%
476   \else%

```

Text before the `\end`, write it with a warning.

```
477     \latex@warning{Writing text ‘##1’ before %
478     \string\end{\@currentenv}\MessageBreak as last line of #1}%
479     \immediate\write\reserved@c{##1}%
480     \fi%
481     \ifx\relax##2\relax%
482     \else%
```

Text after the `\end`, ignore it with a warning.

```
483     \latex@warning{%
484     Ignoring text ‘##2’ after \string\end{\@currentenv}}%
485     \fi%
486     \fi%
487     ^^M}%
```

```
488 \catcode'\^^L\active%
489 \let\L\@undefined%
490 \def^^L{\@ifundefined L^^J^^J^^J}%
491 \catcode'\^^I\active%
492 \let\I\@undefined%
493 \def^^I{\@ifundefined I\space\space}%
494 \catcode'\^^M\active%
495 \edef^^M##1^^M{%
496 \noexpand\reserved@b##1\E\E\relax}}%
497 \endgroup%
```

```
498 \begingroup
499 \catcode'\|=\catcode'\%
500 \catcode'\%=12
501 \catcode'\*=11
502 \gdef\@percentchar{%}
503 \gdef\endfilecontents{|
504 \immediate\closeout\reserved@c
505 \def\T##1##2##3{|
506 \ifx##1\@undefined\else
507 \latex@warning@no@line{##2 has been converted to Blank ##3e}|
508 \fi}|
509 \T\L{Form Feed}{Lin}|
510 \T\I{Tab}{Spac}|
511 \immediate\write\@unused{}}
512 \global\let\endfilecontents*\endfilecontents
513 \@onlypreamble\filecontents
514 \@onlypreamble\endfilecontents
515 \@onlypreamble\filecontents*
516 \@onlypreamble\endfilecontents*
517 \endgroup
518 \@onlypreamble\filec@ntents
```

```
519 </2kernel)
```

69 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```
520 <*afterpreamble>
521 \NeedsTeXFormat{LaTeX2e}
522 \ProvidesPackage{pkgindoc}
523     [1994/10/20 v1.1 Package Interface in Document (DPC)]
524 \def\reserved@a#1\do\@classoptionslist#2\do\filecontents#3\relax{%
525     \gdef\@preamblecmds{#1#3}}
526 \expandafter\reserved@a\@preamblecmds\relax
527 </afterpreamble>
```


File M

lthyphen.dtx

This file contains the code for loading hyphenation patterns into L^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 N

l^tluatex.dtx

70 Overview

Lua_T_EX adds a number of engine-specific functions to _T_EX. Several of these require set up that is best done in the kernel or need related support functions. This file provides *basic* support for Lua_T_EX at the L^A_T_EX 2_ε kernel level plus as a loadable file which can be used with plain _T_EX and L^A_T_EX.

This file contains code for both _T_EX (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 Xe_T_EX, with code defined in `ltfinal.dtx`). With any L^A_T_EX 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^A_T_EX 2_ε kernel did not provide any functionality for the extended allocation area).

71 Core _T_EX functionality

The commands defined here are defined for possible inclusion in a future L^A_T_EX format, however also extracted to the file `ltluatex.tex` which may be used with older L^A_T_EX formats, and with plain _T_EX.

```
\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.

72 Plain T_EX interface

The `luatex` interface may be used with plain T_EX using `\input{luatex}`. This inputs `luatex.tex` which inputs `etex.src` (or `etex.sty` if used with L^AT_EX) if it is not already input, and then defines some internal commands to allow the `luatex` interface to be defined.

The `luatexbase` package interface may also be used in plain T_EX, as before, by inputting the package `\input luatexbase.sty`. The new version of `luatexbase` is based on this `luatex` code but implements a compatibility layer providing the interface of the original package.

73 Lua functionality

73.1 Allocators in Lua

<code>new_attribute</code>	<code>luatexbase.new_attribute(⟨attribute⟩)</code> Returns an allocation number for the <code>⟨attribute⟩</code> , indexed from 1. The attribute will be initialised with the marker value <code>-"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.

73.2 Lua access to T_EX register numbers

<code>registernumber</code>	<code>luatexbase.registernumber(⟨name⟩)</code>
-----------------------------	--

Sometimes (notably in the case of Lua attributes) it is necessary to access a register *by number* that has been allocated by T_EX. This package provides a function to look up the relevant number using LuaT_EX's internal tables. After for example `\newattribute\myattrib`, `\myattrib` would be defined by (say) `\myattrib=\attribute15`. `luatexbase.registernumer("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.registernumer("#1") or "bad input")}%
}

\test{undefinedrubbish}

\test{space}

\test{hbox}

\test{@MM}

\test{@tempdima}
\test{@tempdimb}

\test{strutbox}

\test{sixt@@n}

\attributedef\myattr=12
\myattr=200
\test{myattr}
```

If the demonstration code is processed with LuaL^AT_EX then the following would be produced in the log and terminal output.

```
undefinedrubbish: \relax
    bad input
space: macro:->
    bad input
hbox: \hbox
    bad input
@MM: \mathchar"4E20
    20000
@tempdima: \dimen14
    14
@tempdimb: \dimen15
    15
strutbox: \char"B
    11
sixt@@n: \char"10
```

```

16
myattr: \attribute12
12

```

Notice how undefined commands, or commands unrelated to registers do not produce an error, just return `false` and so print `bad input` here. Note also that commands defined by `\newbox` work and return the number of the box register even though the actual command holding this number is a `\chardef` defined token (there is no `\boxdef`).

73.3 Module utilities

`provides_module` `luatexbase.provides_module(<info>)`
This function is used by modules to identify themselves; the `info` should be a table containing information about the module. The required field `name` must contain the name of the module. It is recommended to provide a field `date` in the usual L^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.

73.4 Callback management

`add_to_callback` `luatexbase.add_to_callback(<callback>, <function>, <description>)` Registers the `<function>` into the `<callback>` with a textual `<description>` of the function. Functions are inserted into the callback in the order loaded.

`remove_from_callback` `luatexbase.remove_from_callback(<callback>, <description>)` Removes the callback function with `<description>` from the `<callback>`. The removed function and its description are returned as the results of this function.

`in_callback` `luatexbase.in_callback(<callback>, <description>)` Checks if the `<description>` matches one of the functions added to the list for the `<callback>`, returning a boolean value.

`disable_callback` `luatexbase.disable_callback(<callback>)` Sets the `<callback>` to `false` as described in the Lua_TE_X manual for the underlying `callback.register` built-in. Callbacks will only be set to `false` (and thus be skipped entirely) if there are no functions registered using the callback.

`callback_descriptions` A list of the descriptions of functions registered to the specified callback is returned. `{}` is returned if there are no functions registered.

`create_callback` `luatexbase.create_callback(<name>, metatype, <default>)` Defines a user defined callback. The last argument is a default function or `false`.

`call_callback` `luatexbase.call_callback(<name>, ...)` Calls a user defined callback with the supplied arguments.

74 Implementation

```
1 (*2ekernel | tex | latexrelease)
2 (2ekernel | latexrelease)\ifx\directlua\@undefined\else
```

74.1 Minimum LuaTeX version

LuaTeX has changed a lot over time. In the kernel support for ancient versions is not provided: trying to build a format with a very old binary therefore gives some information in the log and loading stops. The cut-off selected here relates to the tree-searching behaviour of `require()`: from version 0.60, LuaTeX will correctly find Lua files in the `texmf` tree without ‘help’.

```
3 (latexrelease)\IncludeInRelease{2015/10/01}
4 (latexrelease)          {\newluafunction}{LuaTeX}%
5 \ifnum\luatexversion<60 %
6   \wlog{*****}
7   \wlog{* LuaTeX version too old for ltuatex support *}
8   \wlog{*****}
9   \expandafter\endinput
10 \fi
```

74.2 Older L^AT_EX/Plain T_EX setup

```
11 (*tex)
```

Older L^AT_EX formats don’t have the primitives with ‘native’ names: sort that out. If they already exist this will still be safe.

```
12 \directlua{tex.enableprimitives("",tex.extraprimitives("luatex"))}
13 \ifx\@alloc\@undefined
```

In pre-2014 L^AT_EX, 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
```

74.2.1 Fixes to `etex.src/etex.sty`

These could and probably should be made directly in an update to `etex.src` which already has some LuaTeX-specific code, but does not define the correct range for LuaTeX.

```
27 % 2015-07-13 higher range in luatex
28 \edef \et@xmaxregs {\ifx\directlua\@undefined 32768\else 65536\fi}
29 % luatex/xetex also allow more math fam
30 \edef \et@xmaxfam {\ifx\Umathchar\@undefined\sixt@@n\else\@cclvi\fi}
```

```

31 \count 270=\et@xmaxregs % locally allocates \count registers
32 \count 271=\et@xmaxregs % ditto for \dimen registers
33 \count 272=\et@xmaxregs % ditto for \skip registers
34 \count 273=\et@xmaxregs % ditto for \muskip registers
35 \count 274=\et@xmaxregs % ditto for \box registers
36 \count 275=\et@xmaxregs % ditto for \toks registers
37 \count 276=\et@xmaxregs % ditto for \marks classes

    and 256 or 16 fam. (Done above due to plain/LATEX differences in lAuatex.)
38 % \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}

    End of proposed changes to etex.src

```

74.2.2 luatex specific settings

Switch to global cf `luatex.sty` to leave room for inserts not really needed for `luatex` but possibly most compatible with existing use.

```

39 \expandafter\let\csname newcount\expandafter\expandafter\endcsname
40     \csname globcount\endcsname
41 \expandafter\let\csname newdimen\expandafter\expandafter\endcsname
42     \csname globdimen\endcsname
43 \expandafter\let\csname newskip\expandafter\expandafter\endcsname
44     \csname globskip\endcsname
45 \expandafter\let\csname newbox\expandafter\expandafter\endcsname
46     \csname globbox\endcsname

```

Define `\e@alloc` as in latex (the existing macros in `etex.src` hard to extend to further register types as they assume specific 26x and 27x count range. For compatibility the existing register allocation is not changed.

```

47 \chardef\e@alloc@top=65535
48 \let\e@alloc@chardef\chardef

49 \def\e@alloc#1#2#3#4#5#6{%
50   \global\advance#3\@ne
51   \e@ch@ck{#3}{#4}{#5}#1%
52   \allocationnumber#3\relax
53   \global#2#6\allocationnumber
54   \wlog{\string#6=\string#1\the\allocationnumber}}%

55 \gdef\e@ch@ck#1#2#3#4{%
56   \ifnum#1<#2\else
57     \ifnum#1=#2\relax
58       #1\@cclvi
59       \ifx\count#4\advance#1 10 \fi
60     \fi
61   \ifnum#1<#3\relax
62     \else
63       \errmessage{No room for a new \string#4}%
64     \fi
65   \fi}%

```

Two simple L^AT_EX macros used in `lAlAtex.sty`.

```

66 \long\def\@gobble#1{}
67 \long\def\@firstofone#1{#1}

68 % Fix up allocations not to clash with |etex.src|.

```



```

69 \expandafter\csname newcount\endcsname\@alloc@attribute@count
70 \expandafter\csname newcount\endcsname\@alloc@ccodetable@count
71 \expandafter\csname newcount\endcsname\@alloc@luafunction@count
72 \expandafter\csname newcount\endcsname\@alloc@whatsit@count
73 \expandafter\csname newcount\endcsname\@alloc@bytecode@count
74 \expandafter\csname newcount\endcsname\@alloc@luachunk@count

    End of conditional setup for plain TEX / old LATEX.

75 \fi
76 </tex>

```

74.3 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.

```

77 \ifx\@alloc@attribute@count\@undefined
78   \countdef\@alloc@attribute@count=258
79 \fi
80 \def\newattribute#1{%
81   \@alloc\attribute\attributedef
82   \@alloc@attribute@count\m@ne\@alloc@top#1%
83 }
84 \@alloc@attribute@count=\z@

```

`\setattribute` Handy utilities.
`\unsetattribute`

```

85 \def\setattribute#1#2{#1=\numexpr#2\relax}
86 \def\unsetattribute#1{#1=-"7FFFFFFF\relax}

```

74.4 Category code tables

`\newcatcodetable` Category code tables are allocated with a limit half of that used by LuaT_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.

```

87 \ifx\@alloc@ccodetable@count\@undefined
88   \countdef\@alloc@ccodetable@count=259
89 \fi
90 \def\newcatcodetable#1{%
91   \@alloc\catcodetable\chardef
92   \@alloc@ccodetable@count\m@ne{"8000}#1%
93   \initcatcodetable\allocationnumber
94 }
95 \@alloc@ccodetable@count=\z@

```

`\catcodetable@initex` Save a small set of standard tables. The Unicode data is read here in 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`

```

96 \newcatcodetable\catcodetable@initex
97 \newcatcodetable\catcodetable@string
98 \begingroup
99   \def\setrangecatcode#1#2#3{%

```

```

100 \ifnum#1>#2 %
101   \expandafter\@gobble
102 \else
103   \expandafter\@firstofone
104 \fi
105   {%
106     \catcode#1=#3 %
107     \expandafter\setrange\catcode\expandafter
108     {\number\numexpr#1 + 1\relax}{#2}{#3}
109   }%
110 }
111 \@firstofone{%
112   \catcodetable\catcodetable@initex
113   \catcode0=12 %
114   \catcode13=12 %
115   \catcode37=12 %
116   \setrange\catcode{65}{90}{12}%
117   \setrange\catcode{97}{122}{12}%
118   \catcode92=12 %
119   \catcode127=12 %
120   \save\catcodetable\catcodetable@string
121 \endgroup
122 }%
123 \new\catcodetable\catcodetable@latex
124 \new\catcodetable\catcodetable@atletter
125 \begin\group
126 \def\parseunicodedataI#1;#2;#3;#4\relax{%
127   \parseunicodedataII#1;#3;#2 First>\relax
128 }%
129 \def\parseunicodedataII#1;#2;#3 First>#4\relax{%
130   \ifx\relax#4\relax
131     \expandafter\parseunicodedataIII
132   \else
133     \expandafter\parseunicodedataIV
134   \fi
135   {#1}#2\relax%
136 }%
137 \def\parseunicodedataIII#1#2#3\relax{%
138   \ifnum 0%
139     \if L#21\fi
140     \if M#21\fi
141     >0 %
142     \catcode"#1=11 %
143   \fi
144 }%
145 \def\parseunicodedataIV#1#2#3\relax{%
146   \read\unicoderead to \unicodedataline
147   \if L#2%
148     \count0="#1 %
149     \expandafter\parseunicodedataV\unicodedataline\relax
150   \fi
151 }%
152 \def\parseunicodedataV#1;#2\relax{%
153   \loop

```

```

154     \unless\ifnum\count0>"#1 %
155     \catcode\count0=11 %
156     \advance\count0 by 1 %
157     \repeat
158 }%
159 \def\storedpar{\par}%
160 \chardef\unicoderead=\numexpr\count16 + 1\relax
161 \openin\unicoderead=UnicodeData.txt %
162 \loop\unless\ifeof\unicoderead %
163     \read\unicoderead to \unicodedataline
164     \unless\ifx\unicodedataline\storedpar
165         \expandafter\parseunicodedataI\unicodedataline\relax
166     \fi
167 \repeat
168 \closein\unicoderead
169 \@firstofone{%
170     \catcode64=12 %
171     \savecatcodetable\catcodetable@latex
172     \catcode64=11 %
173     \savecatcodetable\catcodetable@atletter
174 }
175 \endgroup

```

74.5 Named Lua functions

`\newluafunction` Much the same story for allocating LuaTeX 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.

```

176 \ifx\e@alloc@luafunction@count\@undefined
177 \countdef\e@alloc@luafunction@count=260
178 \fi
179 \def\newluafunction{%
180     \e@alloc@luafunction\e@alloc@chardef
181     \e@alloc@luafunction@count\m@ne\e@alloc@top
182 }
183 \e@alloc@luafunction@count=\z@

```

74.6 Custom whatsits

`\newwhatsit` These are only settable from Lua but for consistency are definable here.

```

184 \ifx\e@alloc@whatsit@count\@undefined
185 \countdef\e@alloc@whatsit@count=261
186 \fi
187 \def\newwhatsit#1{%
188     \e@alloc@whatsit\e@alloc@chardef
189     \e@alloc@whatsit@count\m@ne\e@alloc@top#1%
190 }
191 \e@alloc@whatsit@count=\z@

```

74.7 Lua bytecode registers

`\newluabytecode` These are only settable from Lua but for consistency are definable here.

```

192 \ifx\e@alloc@bytecode@count\@undefined
193 \countdef\e@alloc@bytecode@count=262
194 \fi
195 \def\newluabytecode#1{%
196   \e@alloc\luabytecode\e@alloc@chardef
197   \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
198 }
199 \e@alloc@bytecode@count=\z@

```

74.8 Lua chunk registers

`\newluachunkname` As for bytecode registers, but in addition we need to add a string to the `lua.name` table to use in stack tracing. We use the name of the command passed to the allocator, with no backslash.

```

200 \ifx\e@alloc@luachunk@count\@undefined
201 \countdef\e@alloc@luachunk@count=263
202 \fi
203 \def\newluachunkname#1{%
204   \e@alloc\luachunk\e@alloc@chardef
205   \e@alloc@luachunk@count\m@ne\e@alloc@top#1%
206   {\escapechar\m@ne
207   \directlua{lua.name[\the\allocationnumber]="\string#1"}}%
208 }
209 \e@alloc@luachunk@count=\z@

```

74.9 Lua loader

Load the Lua code at the start of every job. For the conversion of \TeX into numbers at the Lua side we need some known registers: for convenience we use a set of systematic names, which means using a group around the Lua loader.

```

210 <2kernel>\everyjob\expandafter{%
211 <2kernel> \the\everyjob
212 \begingroup
213   \attributedef\attributezero=0 %
214   \chardef \charzero =0 %

```

Note name change required on older luatex, for hash table access.

```

215   \countdef \CountZero =0 %
216   \dimendef \dimenzero =0 %
217   \mathchardef \mathcharzero =0 %
218   \muskipdef \muskipzero =0 %
219   \skipdef \skipzero =0 %
220   \toksdef \tokszero =0 %
221   \directlua{require("ltnuatex")}
222 \endgroup
223 <2kernel>}
224 <latexrelease>\EndIncludeInRelease

225 % \changes{v1.0b}{2015/10/02}{Fix backing out of \TeX{ } code}
226 % \changes{v1.0c}{2015/10/02}{Allow backing out of Lua code}
227 <latexrelease>\IncludeInRelease{0000/00/00}
228 <latexrelease> \newluafunction{LuaTeX}%
229 <latexrelease>\let\e@alloc@attribute@count\@undefined
230 <latexrelease>\let\newattribute\@undefined

```

```

231 (latexrelease)\let\setattribute\@undefined
232 (latexrelease)\let\unsetattribute\@undefined
233 (latexrelease)\let\@alloc@ccodetable@count\@undefined
234 (latexrelease)\let\newcatcodetable\@undefined
235 (latexrelease)\let\catcodetable@initex\@undefined
236 (latexrelease)\let\catcodetable@string\@undefined
237 (latexrelease)\let\catcodetable@latex\@undefined
238 (latexrelease)\let\catcodetable@atletter\@undefined
239 (latexrelease)\let\@alloc@luafunction@count\@undefined
240 (latexrelease)\let\newluafunction\@undefined
241 (latexrelease)\let\@alloc@luafunction@count\@undefined
242 (latexrelease)\let\newwhatsit\@undefined
243 (latexrelease)\let\@alloc@whatsit@count\@undefined
244 (latexrelease)\let\newluabytecode\@undefined
245 (latexrelease)\let\@alloc@bytecode@count\@undefined
246 (latexrelease)\let\newluachunkname\@undefined
247 (latexrelease)\let\@alloc@luachunk@count\@undefined
248 (latexrelease)\directlua{luatexbase.uninstall()}
249 (latexrelease)\EndIncludeInRelease

250 (2kernel | latexrelease)\fi
251 (/2kernel | tex | latexrelease)

```

74.10 Lua module preliminaries

```
252 (*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.

```

253 luatexbase      = luatexbase or { }
254 local luatexbase = luatexbase

```

Some Lua best practice: use local versions of functions where possible.

```

255 local string_gsub      = string.gsub
256 local tex_count       = tex.count
257 local tex_setattribute = tex.setattribute
258 local tex_setcount    = tex.setcount
259 local texio_write_nl  = texio.write_nl

```

```

260 local luatexbase_warning
261 local luatexbase_error

```

74.11 Lua module utilities

74.11.1 Module tracking

`modules` To allow tracking of module usage, a structure is provided to store information and to return it.

```
262 local modules = modules or { }
```

`provides_module` Local function to write to the log.

```
263 local function luatexbase_log(text)
```

```

264 texio_write_nl("log", text)
265 end

```

Modelled on `\ProvidesPackage`, we store much the same information but with a little more structure.

```

266 local function provides_module(info)
267   if not (info and info.name) then
268     luatexbase_error("Missing module name for provides_module")
269   end
270   local function spaced(text)
271     return text and (" " .. text) or ""
272   end
273   luatexbase_log(
274     "Lua module: " .. info.name
275     .. spaced(info.date)
276     .. spaced(info.version)
277     .. spaced(info.description)
278   )
279   modules[info.name] = info
280 end
281 luatexbase.provides_module = provides_module

```

74.11.2 Module messages

There are various warnings and errors that need to be given. For warnings we can get exactly the same formatting as from \TeX . For errors we have to make some changes. Here we give the text of the error in the \LaTeX format then force an error from Lua to halt the run. Splitting the message text is done using `\n` which takes the place of `\MessageBreak`.

First an auxiliary for the formatting: this measures up the message leader so we always get the correct indent.

```

282 local function msg_format(mod, msg_type, text)
283   local leader = ""
284   local cont
285   local first_head
286   if mod == "LaTeX" then
287     cont = string_gsub(leader, ".", " ")
288     first_head = leader .. "LaTeX: "
289   else
290     first_head = leader .. "Module " .. msg_type
291     cont = "(" .. mod .. ")"
292     .. string_gsub(first_head, ".", " ")
293     first_head = leader .. "Module " .. mod .. " " .. msg_type .. ":"
294   end
295   if msg_type == "Error" then
296     first_head = "\n" .. first_head
297   end
298   if string.sub(text,-1) ~= "\n" then
299     text = text .. " "
300   end
301   return first_head .. " "
302     .. string_gsub(
303       text
304       .. "on input line "

```

```

305         .. tex.inputlineno, "\n", "\n" .. cont .. " "
306     )
307     .. "\n"
308 end

module\_info Write messages.
module\_warning 309 local function module_info(mod, text)
module\_error 310 texio_write_nl("log", msg_format(mod, "Info", text))
311 end
312 luatexbase.module_info = module_info
313 local function module_warning(mod, text)
314 texio_write_nl("term and log",msg_format(mod, "Warning", text))
315 end
316 luatexbase.module_warning = module_warning
317 local function module_error(mod, text)
318 error(msg_format(mod, "Error", text))
319 end
320 luatexbase.module_error = module_error

Dedicated versions for the rest of the code here.
321 function luatexbase_warning(text)
322 module_warning("luatexbase", text)
323 end
324 function luatexbase_error(text)
325 module_error("luatexbase", text)
326 end

```

74.12 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.

```

327 local luaregisterbasetable = { }
328 local registermap = {
329   attributezero = "assign_attr"   ,
330   charzero      = "char_given"    ,
331   CountZero     = "assign_int"    ,
332   dimenzero     = "assign_dimen"  ,
333   mathcharzero  = "math_given"   ,
334   muskipzero    = "assign_mu_skip",
335   skipzero      = "assign_skip"   ,
336   tokszero     = "assign_toks"   ,
337 }
338 local createtoken
339 if tex.luatexversion > 81 then
340   createtoken = token.create
341 elseif tex.luatexversion > 79 then
342   createtoken = newtoken.create
343 end
344 local hashtokens = tex.hashtokens()
345 local luatexversion = tex.luatexversion
346 for i,j in pairs (registermap) do
347   if luatexversion < 80 then
348     luaregisterbasetable[hashtokens[i][1]] =

```

```

349     hashtokens[i][2]
350   else
351     luaregisterbasetable[j] = createtoken(i).mode
352   end
353 end

```

registernumber Working out the correct return value can be done in two ways. For older LuaTeX releases it has to be extracted from the `hashtokens`. On the other hand, newer LuaTeX's have `newtoken`, and whilst `.mode` isn't currently documented, Hans Hagen pointed to this approach so we should be OK.

```

354 local registernumber
355 if luatexversion < 80 then
356   function registernumber(name)
357     local nt = hashtokens[name]
358     if(nt and luaregisterbasetable[nt[1]]) then
359       return nt[2] - luaregisterbasetable[nt[1]]
360     else
361       return false
362     end
363   end
364 else
365   function registernumber(name)
366     local nt = createtoken(name)
367     if(luaregisterbasetable[nt.cmdname]) then
368       return nt.mode - luaregisterbasetable[nt.cmdname]
369     else
370       return false
371     end
372   end
373 end
374 luatexbase.registernumber = registernumber

```

74.13 Attribute allocation

new_attribute As attributes are used for Lua manipulations its useful to be able to assign from this end.

```

375 local attributes=setmetatable(
376 {},
377 {
378   __index = function(t,key)
379     return registernumber(key) or nil
380   end}
381 )
382 luatexbase.attributes=attributes
383 local function new_attribute(name)
384   tex_setcount("global", "e@alloc@attribute@count",
385               tex_count["e@alloc@attribute@count"] + 1)
386   if tex_count["e@alloc@attribute@count"] > 65534 then
387     luatexbase_error("No room for a new \\attribute")
388   end
389   attributes[name]= tex_count["e@alloc@attribute@count"]
390   luatexbase_log("Lua-only attribute " .. name .. " = " ..
391                 tex_count["e@alloc@attribute@count"])

```



```

392 return tex_count["e@alloc@attribute@count"]
393 end
394 luatexbase.new_attribute = new_attribute

```

74.14 Custom whatsit allocation

`new_whatsit` Much the same as for attribute allocation in Lua.

```

395 local function new_whatsit(name)
396   tex_setcount("global", "e@alloc@whatsit@count",
397               tex_count["e@alloc@whatsit@count"] + 1)
398   if tex_count["e@alloc@whatsit@count"] > 65534 then
399     luatexbase_error("No room for a new custom whatsit")
400   end
401   luatexbase_log("Custom whatsit " .. (name or "") .. " = " ..
402               tex_count["e@alloc@whatsit@count"])
403   return tex_count["e@alloc@whatsit@count"]
404 end
405 luatexbase.new_whatsit = new_whatsit

```

74.15 Bytecode register allocation

`new_bytecode` Much the same as for attribute allocation in Lua. The optional $\langle name \rangle$ argument is used in the log if given.

```

406 local function new_bytecode(name)
407   tex_setcount("global", "e@alloc@bytecode@count",
408               tex_count["e@alloc@bytecode@count"] + 1)
409   if tex_count["e@alloc@bytecode@count"] > 65534 then
410     luatexbase_error("No room for a new bytecode register")
411   end
412   luatexbase_log("Lua bytecode " .. (name or "") .. " = " ..
413               tex_count["e@alloc@bytecode@count"])
414   return tex_count["e@alloc@bytecode@count"]
415 end
416 luatexbase.new_bytecode = new_bytecode

```

74.16 Lua chunk name allocation

`new_chunkname` As for bytecode registers but also store the name in the `lua.name` table.

```

417 local function new_chunkname(name)
418   tex_setcount("global", "e@alloc@luachunk@count",
419               tex_count["e@alloc@luachunk@count"] + 1)
420   local chunkname_count = tex_count["e@alloc@luachunk@count"]
421   chunkname_count = chunkname_count + 1
422   if chunkname_count > 65534 then
423     luatexbase_error("No room for a new chunkname")
424   end
425   lua.name[chunkname_count]=name
426   luatexbase_log("Lua chunkname " .. (name or "") .. " = " ..
427               chunkname_count .. "\n")
428   return chunkname_count
429 end
430 luatexbase.new_chunkname = new_chunkname

```

74.17 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.

74.17.1 Housekeeping

The main table: keys are callback names, and values are the associated lists of functions. More precisely, the entries in the list are tables holding the actual function as `func` and the identifying description as `description`. Only callbacks with a non-empty list of functions have an entry in this list.

```
431 local callbacklist = callbacklist or { }
```

Numerical codes for callback types, and name-to-value association (the table keys are strings, the values are numbers).

```
432 local list, data, exclusive, simple = 1, 2, 3, 4
433 local types = {
434   list      = list,
435   data      = data,
436   exclusive = exclusive,
437   simple    = simple,
438 }
```

Now, list all predefined callbacks with their current type, based on the LuaTeX manual version 0.80. A full list of the currently-available callbacks can be obtained using

```
\directlua{
  for i,_ in pairs(callback.list()) do
    texio.write_nl("- " .. i)
  end
}
```

in plain LuaTeX. (Some undocumented callbacks are omitted as they are to be removed.)

```
439 local callbacktypes = callbacktypes or {
```

Section 4.1.1: file discovery callbacks.

```
440   find_read_file      = exclusive,
441   find_write_file     = exclusive,
442   find_font_file      = data,
443   find_output_file    = data,
444   find_format_file    = data,
445   find_vf_file        = data,
446   find_map_file       = data,
447   find_enc_file       = data,
448   find_sfd_file       = data,
449   find_pk_file        = data,
450   find_data_file      = data,
451   find_opentype_file  = data,
452   find_truetype_file  = data,
453   find_type1_file     = data,
454   find_image_file     = data,
```

Section 4.1.2: file reading callbacks.

```
455 open_read_file      = exclusive,  
456 read_font_file      = exclusive,  
457 read_vf_file        = exclusive,  
458 read_map_file       = exclusive,  
459 read_enc_file       = exclusive,  
460 read_sfd_file       = exclusive,  
461 read_pk_file        = exclusive,  
462 read_data_file      = exclusive,  
463 read_truetype_file  = exclusive,  
464 read_type1_file     = exclusive,  
465 read_opentype_file  = exclusive,
```

Not currently used by luatex but included for completeness. may be used by a font handler.

```
466 find_cidmap_file    = data,  
467 read_cidmap_file    = exclusive,
```

Section 4.1.3: data processing callbacks.

```
468 process_input_buffer = data,  
469 process_output_buffer = data,  
470 process_jobname      = data,
```

Section 4.1.4: node list processing callbacks.

```
471 contribute_filter    = simple,  
472 buildpage_filter     = simple,  
473 pre_linebreak_filter = list,  
474 linebreak_filter     = list,  
475 append_to_vlist_filter = list,  
476 post_linebreak_filter = list,  
477 hpack_filter         = list,  
478 vpack_filter         = list,  
479 hpack_quality        = list,  
480 vpack_quality        = list,  
481 pre_output_filter    = list,  
482 process_rule         = list,  
483 hyphenate            = simple,  
484 ligaturing           = simple,  
485 kerning              = simple,  
486 insert_local_par     = simple,  
487 mlist_to_hlist      = list,
```

Section 4.1.5: information reporting callbacks.

```
488 pre_dump             = simple,  
489 start_run            = simple,  
490 stop_run             = simple,  
491 start_page_number    = simple,  
492 stop_page_number     = simple,  
493 show_error_hook      = simple,  
494 show_warning_message = simple,  
495 show_error_message   = simple,  
496 show_lua_error_hook  = simple,  
497 start_file           = simple,  
498 stop_file            = simple,
```

Section 4.1.6: PDF-related callbacks.

```

499 finish_pdffile = data,
500 finish_pdfpage = data,
Section 4.1.7: font-related callbacks.
501 define_font = exclusive,
502 }
503 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.

```

504 local callback_register = callback_register or callback.register
505 function callback.register()
506   luatexbase_error("Attempt to use callback.register() directly\n")
507 end

```

74.17.2 Handlers

The handler function is registered into the callback when the first function is added to this callback's list. Then, when the callback is called, 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.

Handler for `data` callbacks.

```

508 local function data_handler(name)
509   return function(data, ...)
510     for _,i in ipairs(callbacklist[name]) do
511       data = i.func(data,...)
512     end
513     return data
514   end
515 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.

```

516 local function exclusive_handler(name)
517   return function(...)
518     return callbacklist[name][1].func(...)
519   end
520 end

```

Handler for `list` callbacks.

```

521 local function list_handler(name)
522   return function(head, ...)
523     local ret
524     local alltrue = true
525     for _,i in ipairs(callbacklist[name]) do
526       ret = i.func(head, ...)
527       if ret == false then
528         luatexbase_warning(
529           "Function '" .. i.description .. "' returned false\n"
530           .. "in callback '" .. name .. "'")

```

```

531     )
532     break
533 end
534 if ret ~= true then
535     alltrue = false
536     head = ret
537 end
538 end
539 return alltrue and true or head
540 end
541 end

```

Handler for simple callbacks.

```

542 local function simple_handler(name)
543 return function(...)
544     for _,i in ipairs(callbacklist[name]) do
545         i.func(...)
546     end
547 end
548 end

```

Keep a handlers table for indexed access.

```

549 local handlers = {
550     [data]      = data_handler,
551     [exclusive] = exclusive_handler,
552     [list]      = list_handler,
553     [simple]     = simple_handler,
554 }

```

74.17.3 Public functions for callback management

Defining user callbacks perhaps should be in package code, but impacts on `add_to_callback`. If a default function is not required, it may be declared as `false`. First we need a list of user callbacks.

```

555 local user_callbacks_defaults = { }

```

`create_callback` The allocator itself.

```

556 local function create_callback(name, ctype, default)
557 if not name or name == ""
558 or not ctype or ctype == ""
559 then
560     luatexbase_error("Unable to create callback:\n" ..
561                     "valid callback name and type required")
562 end
563 if callbacktypes[name] then
564     luatexbase_error("Unable to create callback '" .. name ..
565                     "':\ncallback type disallowed as name")
566 end
567 if default ~= false and type (default) ~= "function" then
568     luatexbase_error("Unable to create callback '" .. name ..
569                     "':\ndefault is not a function")
570 end
571 user_callbacks_defaults[name] = default
572 callbacktypes[name] = types[ctype]

```

```

573 end
574 luatexbase.create_callback = create_callback

```

`call_callback` Call a user defined callback. First check arguments.

```

575 local function call_callback(name,...)
576   if not name or name == "" then
577     luatexbase_error("Unable to create callback:\n" ..
578                     "valid callback name required")
579   end
580   if user_callbacks_defaults[name] == nil then
581     luatexbase_error("Unable to call callback '" .. name
582                     .. "':\nunknown or empty")
583   end
584   local l = callbacklist[name]
585   local f
586   if not l then
587     f = user_callbacks_defaults[name]
588     if l == false then
589       return nil
590     end
591   else
592     f = handlers[callbacktypes[name]](name)
593   end
594   return f(...)
595 end
596 luatexbase.call_callback=call_callback

```

`add_to_callback` Add a function to a callback. First check arguments.

```

597 local function add_to_callback(name, func, description)
598   if not name or name == "" then
599     luatexbase_error("Unable to register callback:\n" ..
600                     "valid callback name required")
601   end
602   if not callbacktypes[name] or
603     type(func) ~= "function" or
604     not description or
605     description == "" then
606     luatexbase_error(
607       "Unable to register callback.\n\n"
608       .. "Correct usage:\n"
609       .. "add_to_callback(<callback>, <function>, <description>)"
610     )
611   end

```

Then test if this callback is already in use. If not, initialise its list and register the proper handler.

```

612   local l = callbacklist[name]
613   if l == nil then
614     l = { }
615     callbacklist[name] = l

```

If it is not a user defined callback use the primitive callback register.

```

616     if user_callbacks_defaults[name] == nil then
617       callback_register(name, handlers[callbacktypes[name]](name))
618     end

```

```
619 end
```

Actually register the function and give an error if more than one exclusive one is registered.

```
620 local f = {
621   func      = func,
622   description = description,
623 }
624 local priority = #l + 1
625 if callbacktypes[name] == exclusive then
626   if #l == 1 then
627     luatexbase_error(
628       "Cannot add second callback to exclusive function\n'" ..
629       name .. "'")
630   end
631 end
632 table.insert(l, priority, f)
```

Keep user informed.

```
633 luatexbase_log(
634   "Inserting '" .. description .. "' at position "
635   .. priority .. " in '" .. name .. "'")
636 )
637 end
638 luatexbase.add_to_callback = add_to_callback
```

`remove_from_callback` Remove a function from a callback. First check arguments.

```
639 local function remove_from_callback(name, description)
640   if not name or name == "" then
641     luatexbase_error("Unable to remove function from callback:\n" ..
642       "valid callback name required")
643   end
644   if not callbacktypes[name] or
645     not description or
646     description == "" then
647     luatexbase_error(
648       "Unable to remove function from callback.\n\n"
649       .. "Correct usage:\n"
650       .. "remove_from_callback(<callback>, <description>)"
651     )
652   end
653   local l = callbacklist[name]
654   if not l then
655     luatexbase_error(
656       "No callback list for '" .. name .. "'\n")
657   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.

```
658   local index = false
659   for i,j in ipairs(l) do
660     if j.description == description then
661       index = i
662       break
663     end
```

```

664 end
665 if not index then
666   luatexbase_error(
667     "No callback '" .. description .. "' registered for '" ..
668     name .. "'\n")
669 end
670 local cb = l[index]
671 table.remove(l, index)
672 luatexbase_log(
673   "Removing '" .. description .. "' from '" .. name .. "'."
674 )
675 if #l == 0 then
676   callbacklist[name] = nil
677   callback_register(name, nil)
678 end
679 return cb.func,cb.description
680 end
681 luatexbase.remove_from_callback = remove_from_callback

```

`in_callback` Look for a function description in a callback.

```

682 local function in_callback(name, description)
683   if not name
684     or name == ""
685     or not callbacklist[name]
686     or not callbacktypes[name]
687     or not description then
688     return false
689   end
690   for _, i in pairs(callbacklist[name]) do
691     if i.description == description then
692       return true
693     end
694   end
695   return false
696 end
697 luatexbase.in_callback = in_callback

```

`disable_callback` As we subvert the engine interface we need to provide a way to access this functionality.

```

698 local function disable_callback(name)
699   if(callbacklist[name] == nil) then
700     callback_register(name, false)
701   else
702     luatexbase_error("Callback list for '" .. name .. "' not empty")
703   end
704 end
705 luatexbase.disable_callback = disable_callback

```

`callback_descriptions` List the descriptions of functions registered for the given callback.

```

706 local function callback_descriptions (name)
707   local d = {}
708   if not name
709     or name == ""
710     or not callbacklist[name]

```



```

711     or not callbacktypes[name]
712     then
713     return d
714 else
715 for k, i in pairs(callbacklist[name]) do
716     d[k]= i.description
717     end
718 end
719 return d
720 end
721 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!

```

722 local function uninstall()
723     module_info(
724         "luatexbase",
725         "Uninstalling kernel luatexbase code"
726     )
727     callback.register = callback_register
728     luatexbase = nil
729 end
730 luatexbase.uninstall = uninstall

```

```

731  $\langle$ /lua $\rangle$ 

```

Reset the catcode of `@`.

```

732  $\langle$ tex $\rangle$ \catcode'\@=\etacatcode\relax

```

File O

l^tfinal.d^tx

75 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.

75.1 Debugging

By default, L^AT_EX shows statistics:

```
1 (*2ekernel)
2 \tracingstats1
```

75.2 Typesetting parameters

```
\@lowpenalty These are penalties used internally.
\@medpenalty 3 \newcount\@lowpenalty
\@highpenalty 4 \newcount\@medpenalty
5 \newcount\@highpenalty

\newmarks Allocate extended marks types if etex is active. Placed here at the end of the
format to increase compatibility with count allocations in earlier releases.
6 </2ekernel>
7 < *2ekernel | latexrelease >
8 < latexrelease > \IncludeInRelease{2015/01/01}%
9 < latexrelease > {\newmarks}{Extended Allocation}%
10 \ifx\marks\@undefined\else
11 \def\newmarks{%
12 \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\e@alloc@top}
13 \fi
14 </2ekernel | latexrelease >
15 < latexrelease > \EndIncludeInRelease
16 < latexrelease > \IncludeInRelease{0000/00/00}%
17 < latexrelease > {\newmarks}{Extended Allocation}%
18 < latexrelease > \let\newmarks\@undefined
19 < latexrelease > \EndIncludeInRelease
20 < *2ekernel >

\newXeTeXintercharclass Allocate \XeTeXintercharclass types if xetex is active. previously defined in
\XeTeXalloc@intercharclass xetex.ini.
21 </2ekernel>
22 < *2ekernel | latexrelease >
23 < latexrelease > \IncludeInRelease{2015/01/01}%
24 < latexrelease > {\newXeTeXintercharclass}{Extended Allocation}%
Classes allocated 1 to 254 (early 1, 2 and 3 were pre-set for CJK).
25 \ifx\XeTeXcharclass\@undefined
26 \else
```

```

27 \def\newXeTeXintercharclass{%
28 \e@alloc\XeTeXcharclass\chardef\xe@alloc@intercharclass\m@ne\@cclv}
29 \fi
30 </2ekernel | latexrelease>
31 (latexrelease)\EndIncludeInRelease
32 (latexrelease)\IncludeInRelease{0000/00/00}%
33 (latexrelease) {\newXeTeXintercharclass}{Extended Allocation}%
34 (latexrelease) \ifx\XeTeXcharclass\@undefined
35 (latexrelease) \else
36 (latexrelease) \def\xe@alloc@#1#2#3#4#5{\global\advance#1\@ne
37 (latexrelease) \xe@ch@ck#1#4#2%
38 (latexrelease) \allocationnumber#1%
39 (latexrelease) \global#3#5\allocationnumber
40 (latexrelease) \wlog{\string#5=\string#2\the\allocationnumber}}
41 (latexrelease) \def\xe@ch@ck#1#2#3{%
42 (latexrelease) \ifnum#1<#2\else
43 (latexrelease) \errmessage{No room for a new #3}%
44 (latexrelease) \fi}
45 (latexrelease) \def\newXeTeXintercharclass{%
46 (latexrelease) \xe@alloc@\xe@alloc@intercharclass
47 (latexrelease) \XeTeXcharclass\chardef\@cclv}
48 (latexrelease) \fi
49 (latexrelease)\EndIncludeInRelease
50 (*2ekernel | latexrelease)
51 (latexrelease)\IncludeInRelease{2016/02/01}%
52 (latexrelease) {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
53 \ifx\XeTeXcharclass\@undefined
54 \else
55 \countdef\xe@alloc@intercharclass=257
56 \xe@alloc@intercharclass=\z@
57 \fi
58 </2ekernel | latexrelease>
59 (latexrelease)\EndIncludeInRelease
60 (latexrelease)\IncludeInRelease{2015/01/01}%
61 (latexrelease) {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
62 (latexrelease) \ifx\XeTeXcharclass\@undefined
63 (latexrelease) \else
64 (latexrelease) \xe@alloc@intercharclass=\thr@@
65 (latexrelease) \fi
66 (latexrelease)\EndIncludeInRelease
67 (latexrelease)\IncludeInRelease{0000/00/00}%
68 (latexrelease) {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
69 (latexrelease) \ifx\XeTeXcharclass\@undefined
70 (latexrelease) \else
71 (latexrelease) \newcount\xe@alloc@intercharclass
72 (latexrelease) \xe@alloc@intercharclass=\thr@@
73 (latexrelease) \fi
74 (latexrelease)\EndIncludeInRelease
75 (*2ekernel)

```

The default values of the picture and \fbox parameters:

```

76 \unitlength = 1pt
77 \fboxsep = 3pt
78 \fboxrule = .4pt

```

The saved value of T_EX's `\maxdepth`:

```
79 \@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}`

```
80 \vsize = 1000pt
81 \@colroom = \vsize
82 \@colht = \vsize
```

Initialise `\textheight` `\textwidth` and page style, to avoid internal errors if they are not set by the class.

```
83 \textheight=.5\maxdimen
84 \textwidth=\textheight
85 \ps@empty
```

75.3 Lccodes for hyphenation

For 7- and 8-bit engines the assumption of T1 encodings is the basis for the hyphenation patterns. That's not the case for the Unicode engines, where the assumption is engine-native working. The 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`.

```
86 \ifnum 0%
87 \ifx\Umathcode\undefined\else 1\fi
88 \ifx\XeTeXmathcode\undefined\else 1\fi
89 >\z@
90 \message{ Unicode character data,}
91 \input{load-unicode-data}
92 </2ekernel)
93 (latexrelease)\IncludeInRelease{2016/02/01}%
94 (latexrelease) {\XeTeXintercharclasses}{XeTeX character classes}%
95 (latexrelease) \ifx\XeTeXinterchartoks\undefined
96 (latexrelease) \else
97 (latexrelease) \begingroup
98 (latexrelease) \chardef\XeTeXcharclassID = 0 %
99 (latexrelease) \chardef\XeTeXcharclassOP = 0 %
100 (latexrelease) \chardef\XeTeXcharclassCL = 0 %
101 (latexrelease) \chardef\XeTeXcharclassEX = 0 %
102 (latexrelease) \chardef\XeTeXcharclassIS = 0 %
103 (latexrelease) \chardef\XeTeXcharclassNS = 0 %
104 (latexrelease) \chardef\XeTeXcharclassCM = 0 %
105 (latexrelease) \input{load-unicode-xetex-classes}
106 (latexrelease) \endgroup
107 (latexrelease) \global\let\xtxHanGlue\undefined
108 (latexrelease) \global\let\xtxHanSpace\undefined
109 (latexrelease) \global\XeTeXinterchartoks 0 1 = {}
110 (latexrelease) \global\XeTeXinterchartoks 0 2 = {}
```

```

111 <latexrelease> \global\XeTeXinterchartoks 0 3 = {}
112 <latexrelease> \global\XeTeXinterchartoks 1 0 = {}
113 <latexrelease> \global\XeTeXinterchartoks 2 0 = {}
114 <latexrelease> \global\XeTeXinterchartoks 3 0 = {}
115 <latexrelease> \global\XeTeXinterchartoks 1 1 = {}
116 <latexrelease> \global\XeTeXinterchartoks 1 2 = {}
117 <latexrelease> \global\XeTeXinterchartoks 1 3 = {}
118 <latexrelease> \global\XeTeXinterchartoks 2 1 = {}
119 <latexrelease> \global\XeTeXinterchartoks 2 2 = {}
120 <latexrelease> \global\XeTeXinterchartoks 2 3 = {}
121 <latexrelease> \global\XeTeXinterchartoks 3 1 = {}
122 <latexrelease> \global\XeTeXinterchartoks 3 2 = {}
123 <latexrelease> \global\XeTeXinterchartoks 3 3 = {}
124 <latexrelease> \fi
125 <latexrelease>\EndIncludeInRelease
126 <latexrelease>\IncludeInRelease{0000/00/00}%
127 <latexrelease> {\XeTeXintercharclasses}{XeTeX character classes}%
128 <latexrelease> \ifx\XeTeXinterchartoks\undefined
129 <latexrelease> \else
130 <latexrelease> \input{load-unicode-xetex-classes}
131 <latexrelease> \gdef\xtxHanGlue{\hskipOpt plus 0.1em\relax}
132 <latexrelease> \gdef\xtxHanSpace{\hskip0.2em plus 0.2em minus 0.1em\relax}
133 <latexrelease> \global\XeTeXinterchartoks 0 1 = {\xtxHanSpace}
134 <latexrelease> \global\XeTeXinterchartoks 0 2 = {\xtxHanSpace}
135 <latexrelease> \global\XeTeXinterchartoks 0 3 = {\nobreak\xtxHanSpace}
136 <latexrelease> \global\XeTeXinterchartoks 1 0 = {\xtxHanSpace}
137 <latexrelease> \global\XeTeXinterchartoks 2 0 = {\nobreak\xtxHanSpace}
138 <latexrelease> \global\XeTeXinterchartoks 3 0 = {\xtxHanSpace}
139 <latexrelease> \global\XeTeXinterchartoks 1 1 = {\xtxHanGlue}
140 <latexrelease> \global\XeTeXinterchartoks 1 2 = {\xtxHanGlue}
141 <latexrelease> \global\XeTeXinterchartoks 1 3 = {\nobreak\xtxHanGlue}
142 <latexrelease> \global\XeTeXinterchartoks 2 1 = {\nobreak\xtxHanGlue}
143 <latexrelease> \global\XeTeXinterchartoks 2 2 = {\nobreak\xtxHanGlue}
144 <latexrelease> \global\XeTeXinterchartoks 2 3 = {\xtxHanGlue}
145 <latexrelease> \global\XeTeXinterchartoks 3 1 = {\xtxHanGlue}
146 <latexrelease> \global\XeTeXinterchartoks 3 2 = {\xtxHanGlue}
147 <latexrelease> \global\XeTeXinterchartoks 3 3 = {\nobreak\xtxHanGlue}
148 <latexrelease> \fi
149 <latexrelease>\EndIncludeInRelease
150 (*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.

```
151 \lccode'\- ='\- % default hyphen char
```

The alternative is that a “traditional” engine is in use.

```
152 \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.

```

153 \def\reserved@a#1#2{%
154   \@tempcnta#1\relax
155   \@tempcntb#2\relax

```

```

156 \reserved@b
157 }
158 \def\reserved@b{%
159 \ifnum\@tempcnta>\@tempcntb\else
160 \reserved@c\@tempcnta
161 \advance\@tempcnta\@ne
162 \expandafter\reserved@b
163 \fi
164 }

```

Depending on the T_EX version, we might not be allowed to do this for non-ASCII characters.

```

165 \def\reserved@c#1{%
166 \count@=#1\advance\count@ by -"20
167 \uccode#1=\count@
168 \lccode#1=#1
169 }
170 \reserved@a{'\a}'{\z}
171 \ifnum\inputlineno=\m@ne\else
172 \reserved@a{"A0}"{BC}
173 \reserved@a{"E0}"{FF}
174 \fi

```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```

175 \def\reserved@c#1{%
176 \count@=#1\advance\count@ by "20
177 \uccode#1=#1
178 \lccode#1=\count@
179 \sfcode#1=999
180 }
181 \reserved@a{'\A}'{\Z}
182 \ifnum\inputlineno=\m@ne\else
183 \reserved@a{"80}"{9C}
184 \reserved@a{"C0}"{DF}
185 \fi

```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose `uccode` or `lccode` isn't quite what you'd expect.

```

186 \uccode'\^^Y=' \I % dotless i
187 \lccode'\^^Y=' \^^Y % dotless i
188 \uccode'\^^Z=' \J % dotless j, ae in OT1
189 \lccode'\^^Z=' \^^Z % dotless j, ae in OT1
190 \ifnum\inputlineno=\m@ne\else
191 \lccode'\^^9d=' \i % dotted I
192 \uccode'\^^9d=' \^^9d % dotted I
193 \lccode'\^^9e=' \^^9e % d-bar
194 \uccode'\^^9e=' \^^d0 % d-bar
195 \fi

```

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

```

75.4 Hyphenation

The following code will be compiled into the format file. It checks for the existence of `hyphen.cfg` in inputs that file if found. Otherwise it inputs `hyphen.ltx`. Note that these are loaded in *before* the `\catcodes` are set, so local hyphenation files can use 8-bit input.

We try to load the customized hyphenation description file.

```

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

```

75.5 Font loading

Fonts loaded during the formatting process might already have changed the `\font@submax` from `0pt` to something higher. If so, we put out a bold warning.

```

214 % \changes{v1.1c}{2000/08/23}{Fix typo in warning}
215 \ifdim \font@submax >\z@
216   \@font@warning{Size substitutions with differences\MessageBreak
217                 up to \font@submax\space have occurred.\MessageBreak
218                 \MessageBreak
219                 Please check the transcript file
220                 carefully\MessageBreak
221                 and redo the format generation if necessary!
222                 \@gobbletwo}%
223   \errhelp{Only stopped, to give you time to
224           read the above message.}
225   \errmessage{}

```

We reset the macro. Otherwise every user will get a warning on every job.

```

226 \def\font@submax{0pt}
227 \fi

```

75.6 Input encoding

We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
228 \def\reserved@a#1#2{%
229   \@tempcnta#1\relax
230   \@tempcntb#2\relax
231   \reserved@b
232 }
233 \def\reserved@b{%
234   \ifnum\@tempcnta>\@tempcntb\else
235     \reserved@c\@tempcnta
236     \advance\@tempcnta\@ne
237     \expandafter\reserved@b
238   \fi
239 }
```

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.

```
240 \catcode'\ =10
241 \catcode'\#=6
242 \catcode'\$=3
243 \catcode'\%=14
244 \catcode'\&=4
245 \catcode'\\=0
246 \catcode'\^=7
247 \catcode'\_ =8
248 \catcode'\{=1
249 \catcode'\}=2
250 \catcode'\~=13
251 \catcode'\@=11
252 \catcode'\^^I=10
253 \catcode'\^^J=12
254 \catcode'\^^L=13
255 \catcode'\^^M=5
```

Set the ‘other’ catcodes.

```
256 \def\reserved@c#1{\catcode#1=12\relax}
257 \reserved@c{'!}
258 \reserved@c{'"}
259 \reserved@a{'\'}{'\?}
260 \reserved@c{'\[]
261 \reserved@c{'\]}
262 \reserved@c{'\'}
263 \reserved@c{'\|}
```

Set the ‘letter’ catcodes.

```
264 \def\reserved@c#1{\catcode#1=11\relax}
265 \reserved@a{'\A}'\Z}
266 \reserved@a{'\a}'\z}
```

All the characters in the range 0–31 and 127–255 are illegal, *except* `tab` (`^^I`), `nl` (`^^J`), `ff` (`^^L`) and `cr` (`^^M`).

Now allow 8-bit characters, although their use in this way is strongly discouraged. See `inputenc.dtx` for a supported mechanism for 8-bit input.


```

267 \def\reserved@c#1{\catcode#1=15\relax}
268 \reserved@a{0}{'\^H}
269 \reserved@c{'\^K}
270 \reserved@a{'\^N}{31}
271 %\ifnum\inputlineno=\m@ne
272 \catcode"7F=15
273 %\else
274 % \reserved@a{"7F}{FF}
275 %\fi

```

75.7 Lccodes and uccodes

We now again set up the default (T1) uc/lccodes. The lower case characters need their `\uccode` and `\lccode` values set. Some of this is a repeat of the set-up before loading hyphenation files. Depending on the T_EX version, we might not be allowed to do this for non-ASCII characters. For the Unicode engines (XeT_EX and LuaT_EX) 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.

```

276 \ifnum 0%
277 \ifx\Umathcode\@undefined\else 1\fi
278 \ifx\XeTeXmathcode\@undefined\else 1\fi
279 >\z@
280 \else
281 \def\reserved@c#1{%
282 \count@=#1\advance\count@ by -"20
283 \uccode#1=\count@
284 \lccode#1=#1
285 }
286 \reserved@a{'a}'z}
287 \ifnum\inputlineno=\m@ne\else
288 \reserved@a{"A0}{BC}
289 \reserved@a{"E0}{FF}
290 \fi

```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```

291 \def\reserved@c#1{%
292 \count@=#1\advance\count@ by "20
293 \uccode#1=#1
294 \lccode#1=\count@
295 \sfcode#1=999
296 }
297 \reserved@a{'A}'Z}
298 \ifnum\inputlineno=\m@ne\else
299 \reserved@a{"80}{9C}
300 \reserved@a{"C0}{DF}
301 \fi

```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose `uccode` or `lccode` isn't quite what you'd expect.

```

302 \uccode'\^Y='I % dotless i
303 \lccode'\^Y='\^Y % dotless i
304 \uccode'\^Z='J % dotless j, ae in OT1
305 \lccode'\^Z='\^Z % dotless j, ae in OT1

```

```

306 \ifnum\inputlineno=\m@ne\else
307 \lccode'\^^9d='\i % dotted I
308 \uccode'\^^9d='\^^9d % dotted I
309 \lccode'\^^9e='\^^9e % d-bar
310 \uccode'\^^9e='\^^d0 % d-bar
311 \fi

```

Finally here is one that helps hyphenation in the OT1 encoding.

```

312 \lccode'\^^[='\^^[ % oe in OT1
313 \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.

```

314 \DeclareRobustCommand{\MakeUppercase}[1]{%
315     \def\i{I}\def\j{J}%
316     \def\reserved@a##1##2{\let##1##2\reserved@a}%
317     \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b@gobble}%
318     \protected@edef\reserved@a{\uppercase{#1}}%
319     \reserved@a
320 }
321 \DeclareRobustCommand{\MakeLowercase}[1]{%
322     \def\reserved@a##1##2{\let##2##1\reserved@a}%
323     \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b@gobble}%
324     \protected@edef\reserved@a{\lowercase{#1}}%
325     \reserved@a
326 }
327 \def\@uclclist{\oe\OE\o\O\ae\AE
328     \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:

```

329 \protected@edef\MakeUppercase#1{\MakeUppercase{#1}}
330 \protected@edef\MakeLowercase#1{\MakeLowercase{#1}}

```

75.8 Applying Patch files

Between major releases, small patches will be distributed in files `ltpatch.ltx` which must be added at this point.

Patch file code removed.

```

331 %\IfFileExists{ltpatch.ltx}
332 %  {\typeout{=====^^J%
333 %           Applying patch file ltpatch.ltx^^J%
334 %           =====}}
335 %  \def\fmtversion@topatch{unknown}
336 %  \input{ltpatch.ltx}
337 %  \ifx\fmtversion\fmtversion@topatch
338 %    \ifx\patch@level\undefined
339 %      \typeout{^^J^^J^^J%
340 %              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
341 %              !! Patch file 'ltpatch.ltx' not suitable for this^^J%
342 %              !! version of LaTeX.^^J^^J%
343 %              !! Please check if initex found an old patch file:^^J%
344 %              !! --- if so, rename it or delete it, and redo the^^J%
345 %              !! initex run.^^J%
346 %              !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
347 %      \batchmode \@@end
348 %    \else

```

The code below adds the 'patch level' string to the first `\typeout` in the startup banner.

```

349 %      \def\fmtversion@topatch{0}%
350 %      \ifx\fmtversion@topatch\patch@level\else
351 %        \def\reserved@a\typeout##1##2\reserved@a{%
352 %          \typeout{##1 patch level \patch@level}##2}
353 %        \everyjob\expandafter\expandafter\expandafter{%
354 %          \expandafter\reserved@a\the\everyjob\reserved@a}
355 %        \let\reserved@a\relax
356 %        \the\everyjob
357 %      \fi
358 %    \fi
359 %  \else
360 %    \typeout{^^J^^J^^J%
361 %            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
362 %            !! Patch file 'ltpatch.ltx' (for version <\fmtversion@topatch>)^^J%
363 %            !! is not suitable for version <\fmtversion> of LaTeX.^^J^^J%
364 %            !! Please check if initex found an old patch file:^^J%
365 %            !! --- if so, rename it or delete it, and redo the^^J%
366 %            !!      initex run.^^J%
367 %            !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
368 %    \batchmode \@@end
369 %  \fi
370 %  \let\fmtversion@topatch\relax
371 % }{}

```

75.9 Freeing Memory

`\reserved@a` And just to make sure nobody relies on those definitions of `\reserved@b` and friends. These macros are reserved for use in the kernel. *Do not use them as*

general scratch macros.

```
372 \let\reserved@a\@filelist
373 \let\reserved@b=\@undefined
374 \let\reserved@c=\@undefined
375 \let\reserved@d=\@undefined
376 \let\reserved@e=\@undefined
377 \let\reserved@f=\@undefined
```

`\toks`

```
378 \toks0{}
379 \toks2{}
380 \toks4{}
381 \toks6{}
382 \toks8{}
```

`\errhelp` Empty the error help message, which may have some rubbish:

```
383 \errhelp{}
```

75.10 Initialise file list

`\@providesfile` Initialise for use in the document. During `initex` a modified version has been used which leaves debugging information for `latexbug.tex`.

```
384 \def\@providesfile#1[#2]{%
385   \wlog{File: #1 #2}%
386   \expandafter\xdef\csname ver@#1\endcsname{#2}%
387   \endgroup}
```

`\@filelist` Reset `\@filelist` so files input while making the format are not listed. The list built up so far may take up a lot of memory and so it is moved to `\reserved@a` where it will be overwritten as soon as almost any `LATEX` command is issued in a class file. However the `latexbug.tex` program will be able to access this information and insert it into a bug report.

```
388 \let\@filelist\@gobble
389 \def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}%
```

75.11 Dumping the format

Finally we make `@` into a letter, ensure the format will be in the ‘normal’ error mode, and dump everything into the format file.

```
390 \makeatother
391 \errorstopmode
392 \dump
393 </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.	264
1989/04/10 ltfssbas.dtx v1.0a	
General: Starting with version numbers! <code>\ifmmode</code> added in <code>\math@group</code>	139
1989/04/10 ltfssbas.dtx v1.0b	
General: <code>\preload@sizes</code> added. <code>\wrong@fontshape</code> changed to define substitution font/shape macro.	139
1989/04/10 ltfssini.dtx v1.0a	
General: Starting with version numbers <code>\newif</code> for <code>\@tempswa</code> added since this switch is unknown at the time when this file is read in. (latex.tex is loaded later.) <code>\math@famname</code> changed to <code>\math@version</code> .	208
1989/04/14 ltfssbas.dtx v1.0c	
General: More documentation added.	139
1989/04/15 ltfssini.dtx v1.0b	
General: <code>\mathfontset</code> renamed to <code>\mathversion</code> .	208
1989/04/19 ltfssbas.dtx v1.0d	
General: Even more doc.	139
1989/04/21 ltfssbas.dtx v1.0e	
General: Documentation is fun! Parameters of <code>\define@mathalphabet</code> changed.	139
1989/04/21 ltfssini.dtx v1.0c	
General: Changed to conform to fam.tex.	208
1989/04/23 ltfssbas.dtx v1.0f	
General: % in <code>\getanddefinefonts</code> added.	139
1989/04/26 ltfssini.dtx v1.0d	
General: <code>\xpt</code> added.	208
1989/04/27 ltfssbas.dtx v1.0g	
General: Documentation revised.	139
1989/04/27 ltfssini.dtx v1.0e	
General: Definitions of L ^A T _E X symbols corrected.	208
1989/04/29 ltfssbas.dtx v1.0h	
General: Documented problem with <code>\halign</code> , and <code>\noalign</code>	139
<code>\mathversion</code> : Test if version defined added.	147
1989/04/29 ltfssbas.dtx v1.0i	
General: Removed the <code>\halign</code> <code>\noalign</code> correction (wasn't bugfree)	139
1989/04/29 ltfssini.dtx v1.0f	
General: Corrections to L ^A T _E X tabular env. added.	208
1989/05/01 ltfssbas.dtx v1.0j	
General: Default for <code>\baselinestretch</code> added.	139
1989/05/22 ltfssbas.dtx v1.0k	
General: Lines longer than 72 characters folded.	139
1989/05/22 ltfssini.dtx v1.0g	
General: Lines shortened to 72 characters	208
1989/09/14 ltfssbas.dtx v1.0m	
General: Global replacement: <code>\group</code> to <code>\mathgroup</code>	139
<code>\mathversion</code> : Corrected typo: <code>\endscname</code> to <code>\endcsname</code> .	147
1989/11/07 ltfssini.dtx v1.0i	
General: All family, series, and shape names abbreviated.	208
1989/11/08 ltfssbas.dtx v1.0o	
General: First parameter of <code>\define@mathalphabet</code> and <code>\define@mathgroup</code> changed from string to control sequence.	139
1989/11/14 ltfssbas.dtx v1.0p	
<code>\math@version</code> : Math version prefix 'mv@' added.	147
1989/11/19 ltfssbas.dtx v1.0q	
<code>\define@newfont</code> : Group added.	149
<code>\wrong@fontshape</code> : Instead of calling <code>\family\default@family</code> , etc. we directly set <code>\f@family</code> , etc.	152
1989/11/22 ltfssbas.dtx v1.0r	
<code>\math@version</code> : <code>\def</code> → <code>\edef</code> for <code>\math@version</code> .	147
1989/11/25 ltfssbas.dtx v1.0s	
General: All <code>\edef\font@name</code> changed to <code>\xdef\font@name</code> .	

Necessary after introduction of <code>\begin@group/\endgroup</code> in v1.0q.	139	1990/01/21 ltfssstrc.dtx v1.2b <code>\use@mathgroup</code> : Macro added to allow cleaner interface.	169
<code>extra// → +</code> in <code>\extra@def</code>	139	1990/01/23 ltfssbas.dtx v1.2c General: <code>\no@version@warning</code> renamed to <code>\no@alphabet@error</code>	139
1989/11/26 ltfssbas.dtx v1.0t <code>\select@group: \bgroup/\egroup</code> changed to <code>\begin@group/\endgroup</code> to avoid empty Ord atom on math list.	154	Macro <code>\no@alphabet@help</code> added	139
1989/12/02 ltfssini.dtx v1.1b General: <code>\rmmath</code> renamed to <code>\mathrm</code>	208	<code>\no@alphabet@error</code> : Changed to error call	139
1989/12/03 ltfssini.dtx v1.1c General: Some internal macros renamed to make them inaccessible.	208	1990/01/25 ltfssini.dtx v1.1e <code>\nfss@text</code> : Macro added.	211
1989/12/05 ltfssbas.dtx v1.0u <code>\addto@hook: \addto@hook</code> added.	157	1990/01/27 ltfssbas.dtx v1.2d <code>\DeclarePreloadSizes</code> : Font identifier set to <code>\relax</code>	144
1989/12/05 ltfssstrc.dtx v1.0u fam.dtx <code>\every@math@size</code> : Hook <code>\every@size</code> added.	166	1990/01/28 ltfssbas.dtx v1.2e <code>\mathgroup: \newfam</code> let to <code>\new@mathgroup</code>	139
1989/12/13 ltfssstrc.dtx v1.0f <code>\use@mathgroup: \expandafter</code> added before final <code>\fi</code>	169	1990/01/28 ltfssbas.dtx v1.2f <code>\define@newfont</code> : Added call to <code>\curr@fontshape</code> macro to allow substitution.	150
1989/12/16 ltfssbas.dtx v1.1a <code>\select@group: \relax</code> in front added.	154	<code>\wrong@fontshape</code> : Warning message slightly changed.	152
Now four arguments.	154	1990/01/28 ltfssini.dtx v1.2b <code>\em</code> : Call to <code>\@nomath</code> added.	209
Redefinition of alphabet now simpler.	155	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>	208
Usage of ‘=’ macro added.	155	1990/02/16 ltfssbas.dtx v1.2g General: Support for changes of <code>\baselineskip</code> without changing the size.	139
1989/12/16 ltfssstrc.dtx v1.1a <code>\selectfont</code> : Changed order of calls.	163	<code>\math@version: \@nomath</code> added.	147
<code>\use@mathgroup</code> : Redefinition of alphabet now simpler.	169	1990/02/16 ltfssstrc.dtx v1.0i <code>\selectfont</code> : Changed <code>\f@size</code> to <code>\lcl@currsiz</code> (see fam file).	163
Usage of ‘=’ macro added.	169	1990/02/18 ltfssstrc.dtx v1.0j General: Redefine unprotected version <code>\p@selectfont</code> instead of <code>\selectfont</code>	163
1990/01/18 ltfssstrc.dtx v1.0h General: <code>\tracingfonts</code> meaning changed.	159	1990/03/14 ltfssstrc.dtx v1.0k General: Added code for TeX3.	159
1990/01/20 ltfssbas.dtx v1.2a <code>\math@bgroup</code> : Def. placed in this file.	156	<code>\extract@font</code> : Added code for TeX3.	162
<code>\math@egroup</code> : Def. placed in this file.	156	<code>\selectfont</code> : Added code for TeX3.	163
<code>\select@group</code> : Def for alph id changed.	155	1990/03/30 ltfssbas.dtx v1.2h <code>\math@egroup</code> : Changed to have one arg.	156
1990/01/21 ltfssbas.dtx v1.2b <code>\select@group</code> : Code moved to <code>\use@mathgroup</code>	155		

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>). 169	<code>\type@restoreinfo</code> : Some extra tracing info. 165
1990/04/01 ltfssbas.dtx v1.2i	1990/08/27 ltfssstrc.dtx v1.0r
General: Code added from <code>tracefnt.dtx</code> 139	<code>\getanddefine@fonts</code> : Correcting missing name after <code>\tracingon</code> 170
Support for TeX3. 139	1991/03/28 ltfssini.dtx v1.1m
1990/04/01 ltfssstrc.dtx v1.0l	<code>\copyright</code> : Extra braces added. 211
General: Part of code moved to <code>fam.dtx</code> 159	1991/03/30 ltfssini.dtx v1.2g
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> already defined. 160	<code>\newfont</code> : Definition added. 210
1990/04/01 ltfssstrc.dtx v1.0o	<code>\symbol</code> : Definition added. 210
<code>\tracingfonts</code> : Check if <code>\tracingfonts</code> defined removed again. 160	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 255
General: <code>\input</code> of files now handled by <code>docstrip</code> 208	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 261
<code>\selectfont</code> : Call <code>\tracingon</code> only if <code>\tracingfonts</code> greater than 3. 163	1991/08/14 ltpictur.dtx LaTeX2.09
1990/05/05 ltfssstrc.dtx v1.0n	General: (RmS) inserted extra braces around entry for NFSS 321
<code>\selectfont</code> : <code>\tracingon</code> with new syntax. 163	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 343
<code>\nfss@text</code> : Changed to <code>\mbox</code> . . 211	1991/08/26 ltfssini.dtx v1.1n
1990/06/24 ltfssbas.dtx v1.2j	<code>\p@reset@font</code> : Macro introduced 211
<code>\DeclarePreloadSizes</code> : Missing percent added. 143	1991/08/26 ltmiscen.dtx LaTeX2.09
1990/06/24 ltfssstrc.dtx v1.0o	<code>\@verbatim</code> : <code>\@@par</code> added 255
<code>\baselinestretch</code> : Moved to <code>tracefnt.dtx</code> 166	1991/08/26 ltpictur.dtx LaTeX2.09
<code>\getanddefine@fonts</code> : Adding tracing code. 170	<code>\endpicture</code> : (RmS & FmI) extra boxing level around <code>\@picbox</code> to guard against unboxing in math mode (proposed by John Hobby) 320
<code>\Macro</code> moved from <code>fam.dtx</code> . . 170	1991/08/26 ltplain.dtx LaTeX2.09
Adding debug code. 170	<code>\tracingall</code> : Added <code>\errorcontextlines=\maxdimen</code> , suggested by J. Schrod 29
<code>\use@mathgroup</code> : Tracing code added. 169	1991/09/29 ltboxes.dtx LaTeX2.09
1990/06/30 ltfssbas.dtx v1.2l	<code>\@mpfootnotetext</code> : (RmS) added <code>\reset@font</code> 293
<code>\showhyphens</code> : Macro added. . . . 157	1991/09/29 ltfloating.dtx LaTeX2.09
1990/06/30 ltfssstrc.dtx v1.0p	<code>\@footnotetext</code> : (RmS) added <code>\reset@font</code> 372
<code>\use@mathgroup</code> : Added <code>\relax</code> after math group number. . . 169	1991/09/29 ltmath.dtx LaTeX2.09
1990/07/07 ltfssstrc.dtx v1.0q	<code>\@eqnnum</code> : RmS: <code>\reset@font</code> added. 264
<code>\getanddefine@fonts</code> : Group number added to tracing. . . 170	1991/09/29 ltsect.dtx LaTeX2.09
<code>\math@egroup</code> : Tracing code added. 169	<code>\@dottedtocline</code> : (RmS) added <code>\reset@font</code> for page number 353
<code>\use@mathgroup</code> : Group number added to tracing. 169	

- 1991/10/17 ltentrldtx LaTeX209
`\@tfor`: (Rms) `\xdef` replaced by
`\def` (See FMI's array.doc) . . . 54
- 1991/10/25 ltbibldtx LaTeX2.09
`\@citex`: added `\reset@font`,
suggested by Bernd Raichle. 377
- 1991/11/01 ltfloatedtx LaTeX2.09
`\footnote`: (RmS) Added
`\let\protect\noexpand` in
`\footnote`, `\footnotemark`,
and `\footnotetext`, since
`\xdef` is used 371
- 1991/11/04 ltlistsdtx LaTeX2.09
`\makelabel`: (RmS) added default
definition for `\makelabel`, to
produce an error message. . . . 281
- 1991/11/04 ltplaindtx RmS
General: Removed `\itemitem` since
never needed/useful with
 \LaTeX 28
- 1991/11/06 ltbibldtx LaTeX2.09
`\@citex`: added code to remove a
leading blank 377
- 1991/11/13 ltbibldtx LaTeX2.09
`\@bibitem`: Changed counter
`enumi` to `enumiv`, as it says in
the comment above 377
- 1991/11/21 ltfssinidtx v1.1o
`\p@reset@font`: Added extra
braces for robustness. 211
Changed to protected version of
macro. 211
- 1991/11/22 ltfloatedtx LaTeX2.09
`\footnote`: (RmS) Added
`\let\protect\noexpand` in
`\@xfootnote`,
`\@xfootnotemark`, and
`\@xfootnotetext` 371
- 1991/11/22 ltlistsdtx LaTeX2.09
`\@item`: (RmS) Changed second
call to `\makelabel` to
`\unhbox\@tempboxa`. Avoids
problems with side effects in
`\makelabel` and is more
efficient. 281
- 1991/11/27 ltfssbasdtx v1.3a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily` etc. 139
- 1991/11/27 ltfssinidtx v1.2a
General: All `\family`, `\shape` etc.
renamed to `\fontfamily` etc. 208
- 1992/01/06 ltfssinidtx v1.2c
General: added `slitex` code 208
- 1992/01/10 ltbibldtx LaTeX2.09
`\@bibitem`: Changed `\c@enumiv` to
`\value of \@listctr` 377
- 1992/01/10 ltmathdtx LaTeX2.09
`equation`: RmS: put `\hbox` around
`\@eqnnum` to typeset the
equation number in text mode
(as in the `eqnarray` env.) 264
- 1992/01/10 ltthmdtx LaTeX2.09
`\@othm`: (RmS) Check for existence
of theorem environment 342
- 1992/01/14 ltbibldtx LaTeX2.09
`\@biblabel`: removed `\hfill` . . . 379
- 1992/01/14 ltsectdtx 0.0
`\@starttoc`: (RmS) added
`\immediate` to `\openout` as all
`\write` commands are also
executed `\immediate` 352
- 1992/02/26 ltbibldtx LaTeX2.09
`\@lbibitem`: Added `\hfill` to
restore left-alignment of
bibliography labels in alpha
style 377
- 1992/03/18 ltdefnsdtx LaTeX209
General: (RMS) changed input
channel from 0 to
`\inputcheck` to avoid
conflicts with other channels
allocated by `\newread` 36
- 1992/03/18 ltfloatedtx LaTeX2.09
`\@xympar`: (RmS) added
`\global\@ignorefalse` 367
- `\end@float`: (RmS) changed
`\@esphack` to `\@Eshpack` 361
- 1992/03/18 ltlistsdtx 0.0
General: RmS: added
`\@nmbrlistfalse` 278
- 1992/03/18 ltmiscendtx LaTeX2.09
`\begin`: Changed `\@ignoretrue` to
`\@ignorefalse` (as
documented) 253
- 1992/03/21 ltfssinidtx v1.2d
General: Renamed `\text` to
`\nfss@text` to make it
internal. 208
- 1992/05/12 ltfssbasdtx v1.3c
`\extract@alph@from@version`:
Macro added. 155
- `\select@group`: Added call to `\ex-`
`tract@alph@from@version`. . . 155
- 1992/07/26 ltfssbasdtx v1.9a
`\curr@fontshape`: 149
- `\DeclareFontShape`: Introduced
`\DeclareFontShape` 140

<code>\define@newfont:</code>	149	<code>\@seccntformat</code>	347
<code>\math@fonts:</code>	154	1992/09/18 <code>ltlists.dtx</code> LaTeX2.09	
<code>\select@group:</code>	154, 155	General: (RmS) Added warning if	
<code>\split@name:</code> Added splitting into		<code>\item</code> is used in math mode	279
<code>\f@encoding.</code>	149	1992/09/18 <code>lftab.dtx</code> LaTeX2.09	
<code>\wrong@fontshape:</code>	152	<code>\@array:</code> Changed <code>\par</code> to	
1992/07/26 <code>lftsstrc.dtx</code> v2.0b		<code>\@empty</code> to avoid starting new	
<code>\s@fct@:</code>	178	row e.g. after <code>\hline</code>	307
<code>\s@fct@sub:</code>	179	1992/09/19 <code>lftsstrc.dtx</code> v2.0c	
<code>\selectfont:</code>	163	<code>\try@simple@size:</code>	172
<code>\try@simple@size:</code>	172, 173	1992/09/21 <code>lftssini.dtx</code> v1.4d	
<code>\try@size@range:</code>	176	<code>\not@math@alphabet:</code> Macro	
<code>\use@mathgroup:</code>	169	defined.	209
1992/08/14 <code>ltbibl.dtx</code> LaTeX2.09		1992/09/22 <code>lftssbas.dtx</code> v1.91a	
<code>\@citex:</code> added missing argument		General: Introduced <code>\tf@size</code> for	
braces around <code>\hbox</code> , found by		math size.	139
Ed Szynter	377	1992/09/22 <code>lftsstrc.dtx</code> v2.1a	
1992/08/14 <code>ltboxes.dtx</code> LaTeX2.09		<code>\getanddefine@fonts:</code> Introduced	
<code>\endminipage:</code> (RmS) replaced		<code>\tf@size</code> for math size.	170
<code>\vskip-\lastskip</code> by <code>\unskip</code>		1992/11/13 <code>lftssini.dtx</code> v?	
(proposed by FMi)	293	<code>\hexnumber@:</code> Made expandable.	210
1992/08/17 <code>ltbibl.dtx</code> LaTeX2.09		1992/11/23 <code>ltcounts.dtx</code> 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	133
Krab Thorup)	377	1992/11/26 <code>ltboxes.dtx</code> LaTeX2.09	
1992/08/19 <code>ltsect.dtx</code> 0.0		<code>\@mpfootnotetext:</code> (RmS) added	
<code>\@xsect:</code> (RmS) corrected bug:		protection for <code>\edef</code>	293
stretch and shrink in argument		1992/11/26 <code>lftfloat.dtx</code> LaTeX2.09	
to <code>\hskip</code> previously not		<code>\@footnotetext:</code> (RmS) added	
negated	348	protection for <code>\edef</code>	372
1992/08/19 <code>ltthm.dtx</code> 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		371
undefined counter	342	1992/12/03 <code>lftssini.dtx</code> v?	
1992/08/20 <code>lftssini.dtx</code> v1.4b		<code>\hexnumber@:</code> Make it accept	
<code>\@setsize:</code> Added <code>\@currsiz</code>	210	counters.	210
1992/08/24 <code>ltdefns.dtx</code> LaTeX2.09		1993/03/08 <code>preload.dtx</code> v2.0b	
<code>\@ifnextchar:</code> (Rms)		General: Added 12pt preloads	232
<code>\@ifnextchar</code> didn't work if its		1993/03/18 <code>lftssbas.dtx</code> v2.0c	
first argument was an equal		General: Changed all <code>\@tempdima</code>	
sign.	46	in <code>\@tempdimb</code> to avoid killing	
1992/08/24 <code>ltmiscen.dtx</code> LaTeX2.09		<code>\numberline</code>	139
<code>\begin:</code> Added code to <code>\begin</code> to		1993/03/18 <code>lftsstrc.dtx</code> 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>	253	<code>\numberline</code>	159
<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	256	<code>\numberline</code>	159
1992/08/25 <code>ltsect.dtx</code> LaTeX2.09		1993/03/18 <code>lftsstrc.dtx</code> 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	175

1993/04/09 lterror.dtx v1.0e		1993/09/02 ltfstrc.dtx v2.1i	
<code>\@latexerr</code> : Mention The Companion	60	General: Corrected name of sgen size function.	159
1993/04/11 lterror.dtx v1.0f		1993/09/03 ltmiscen.dtx LaTeX2.09	
<code>\@latexerr</code> : Remove setting of errorcontextlines	60	<code>\verbatim@nolig@list</code> : Replaced <code>\noligs</code> by extensible list	256
1993/05/05 ltfntcmd.dtx v2.0b		1993/09/07 ltmiscen.dtx LaTeX2.09	
General: Removed all LaTeX related cmds	236	<code>\verb@balance@group</code> : (RmS) Changed definition of <code>\verb</code> so that it detects a missing second delimiter.	256
1993/05/16 ltfssbas.dtx v2.0e		1993/09/08 ltmiscen.dtx LaTeX2.09	
<code>\showwhyphens</code> : Use <code>\reset@font</code>	157	<code>\enddocument</code> : Added warning in case of undefined references.	250
1993/07/16 ltfstrc.dtx v2.1h		1993/09/15 ltfssbas.dtx v2.0g	
General: Changed layout of info messages	159	<code>\DeclareFontEncoding</code> : Corrected: <code>\default@T</code> to <code>\default@M</code>	142
1993/07/17 ltoutenc.dtx 1.0d		1993/09/15 ltfstrc.dtx v2.1j	
General: changed <code>\catcoding@</code>	94	General: Corrected spelling of <code>\noexpand</code>	159
1993/08/03 ltmiscen.dtx LaTeX2.09		1993/09/19 lterror.dtx LaTeX2.09	
<code>\enddocument</code> : Changed redefinition of <code>\global</code> to redefinition of <code>\@setckpt</code>	250	<code>\@invalidchar</code> : (RmS) Error message for invalid input characters.	62
1993/08/05 ltpictur.dtx LaTeX2.09		1993/11/02 ltmath.dtx LaTeX2.09	
<code>\circle</code> : (RMS) Added error message if <code>\circle</code> is used in math mode.	336	General: RmS: Corrected description of <code>\@eqnset</code> , moved <code>\@eqnset</code> accordingly and removed extra <code>\tabskip</code> assignment.	264
1993/08/05 ltsect.dtx LaTeX2.09		1993/11/03 ltmath.dtx LaTeX2e	
<code>\@sect</code> : (RmS) Made sure that <code>\protect</code> works correctly in expansion of <code>\the counter</code>	347	General: RmS: Initialized <code>\everycr</code> to empty	264
1993/08/05 ltspace.dtx LaTeX2e		1993/11/03 ltpictur.dtx LaTeX2.09	
<code>\hspace</code> : (RmS) Removed superfluous <code>\leavevmode</code> in <code>\@hspace</code> and <code>\@hspacer</code> , as suggested by CAR.	79	General: (RmS) changed <code>\halign</code> to <code>\ialign</code> to initialize <code>\tabskip</code> and <code>\everycr</code>	321
1993/08/05 lttab.dtx latex2e		1993/11/11 ltfssini.dtx v2.1a	
<code>\tabular*</code> : Replaced <code>\expandafter\def</code> by <code>\@namedef</code>	307	<code>\normalfont</code> : Macro added	211
1993/08/06 ltbibl.dtx LaTeX2.09		1993/11/11 ltfstrc.dtx v2.2a	
<code>\@citex</code> : Moved writing to .aux file in loop over citation keys so that leading blanks are removed there as well.	377	General: Option concept added for LaTeX2e	159
1993/08/13 ltoutenc.dtx 1.0f		1993/11/14 ltclass.dtx v0.2a	
General: Protected against active <code>@ sign</code>	94	<code>\@currentx</code> : Name changed from <code>\@currentextension</code>	456
1993/08/13 preload.dtx v2.0c		<code>\@fileswithoptions</code> : Moved resetting of <code>\default@ds</code> , <code>\ds@</code> and <code>\@declaredoptions</code> here, from the end of <code>\ProcessOptions</code>	464
General: Added <code>\relax</code> at end of font names.	233	<code>\@reset@options</code> : macro added	466
1993/08/16 ltoutenc.dtx 1.0g		<code>\AtEndDocument</code> : Included extension in the generated	
General: Needs space after <code>\string</code>	94		
1993/08/18 ltfssdcl.dtx v2.0e			
<code>\new@mathversion</code> : Exchanged names of encodings in warning message of <code>\SetSymbolFont</code>	193		

macro name for package and class hooks.	466	1993/11/22 ltclass.dtx v0.2f	
<code>\documentstyle</code> : Added		<code>\@fileswithoptions</code> : Made the default [] not	
<code>\RequirePackage</code>		<code>[\@unknownversion]</code>	464
<code>\@unusedoptionlist</code> stuff. . .	462	Made the initial version [] not	
<code>\g@addto@macro</code> : Made global . .	466	<code>[\@unknownversion]</code>	464
<code>\NeedsTeXFormat</code> : made more robust for alternative syntax for other formats.	463	<code>\@ifclasslater</code> : Added //00 so parsing never produces a runaway argument.	458
<code>\ProcessOptions*</code> : Optimise 'empty option' code.	460	General: <code>\@unknownversion</code> removed	469
Stop adding the global option list inside class files.	460	1993/11/22 ltdefs.dtx LaTeX2e	
1993/11/15 ltclass.dtx v0.2b		<code>\@minus</code> : Macro added	35
<code>\documentstyle</code> : Modified to match <code>\ProcessOption*</code> . . .	462	<code>\@plus</code> : Macro added	35
<code>\ProcessOptions*</code> : Star form added.	460	<code>\CheckCommand</code> : Macro added . . .	41
1993/11/17 ltclass.dtx v0.2c		<code>\providecommand</code> : Macro added . .	41
<code>\@@fileswith@pti@ns</code> : Macro added	465	1993/11/22 lterror.dtx LaTeX2e	
<code>\@badrequireerror</code> : Macro added	467	<code>\c@errorcontextlines</code> : Macro added	59
<code>\@fileswithoptions</code> : Added trap for two <code>\LoadClass</code> commands.	465	1993/11/22 ltfiles.dtx LaTeX2e	
<code>\@twoloadclasserror</code> : Macro added	467	<code>\listfiles</code> : Removed checking for <code>\@unknownversion</code>	89
<code>\CurrentOption</code> : Name changed from <code>\@curroption</code>	456	1993/11/22 ltlength.dtx LaTeX2e	
<code>\DeclareOption*</code> : Error checking added	459	<code>\@settodim</code> : Macro added	138
<code>\NeedsTeXFormat</code> : Name changed from <code>\NeedsFormat</code>	463	<code>\@settopoint</code> : Macro added	138
<code>\ProcessOptions*</code> : restoring <code>\@fileswith@pti@ns</code> added.	460	<code>\settodepth</code> : Macro added	138
1993/11/18 ltclass.dtx v0.2d		<code>\settoheight</code> : Macro added	138
<code>\documentstyle</code> : Modified		1993/11/22 ltlogos.dtx LaTeX2e	
<code>\RequirePackage</code> stuff.	462	<code>\LaTeXe</code> : Macro added	80
<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>	461
<code>\reserved@a</code>	461	General: Various macros now moved to latex.tex.	456
<code>\NeedsTeXFormat</code> : <code>\fmtname</code>		Warnings and errors now directly coded.	456
<code>\fmtversion</code> not <code>\@...</code>	463	1993/11/23 ltdefs.dtx LaTeX2e	
1993/11/21 ltfiles.dtx LaTeX2e		<code>\@argdef</code> : Macro added	37
<code>\@missingfileerror</code> : Stop infinite looping on <code>\@er@ext</code>	88	<code>\@ifundefined</code> : Redefined to remove a trailing <code>\fi</code>	46
1993/11/21 ltmiscen.dtx v0.9a		<code>\@newcommand</code> : Macro added	37
<code>\@verbatim</code> : use <code>\verbatim@font</code> instead of <code>\tt</code>	255	<code>\@newenv</code> : Macro interface changed	40
<code>\verb</code> : Use <code>\verbatim@font</code> instead of <code>\tt</code>	256	<code>\@xargdef</code> : Macro interface changed	37
<code>\verbatim@font</code> : Macro added . .	256	<code>\@yargdf</code> : Avoid <code>\@?@?</code> token . .	38
		Macro interface changed	38
		<code>\newcommand</code> : Macro reimplemented and extended . .	37
		<code>\renewcommand</code> : Macro reimplemented and extended . .	39
		<code>\renewenvironment</code> : Macro reimplemented and extended . .	40
		<code>\two@digits</code> : Macro added	34

1993/11/23 ltoutput.dtx v0.1a		\@imakebox: macro modified ...	286
\paperheight: Register added ..	396	\@irsbox: redefined to support	
\paperwidth: Register added ...	396	\height	294
1993/11/23 ltoutput.dtx v0.1c		\@isavebox: color support	288
\@enlargepage: Command added	436	extra group	288
\@kludgeins: Insert added	436	\@isavepicbox: extra group	288
\@makecol: Command changed ..	406	\@makebox: default changed from x	
\@specialoutput: Command		to c	286
changed	399	\@makepicbox: macro modified ..	287
\enlargethispage*: Commands		\@savebox: default c not x	288
added	436	\bm@b: macros added	286
1993/11/24 ltfntcmd.dtx v2.1a		\endlrbox: macro added	288
\maybe@ic@: Use \t@st@ic	241	\fbox: extra group	289
\t@st@ic: Macro added	241	\lrbox: color support	288
1993/11/24 ltffsini.dtx v2.1a		macro added	288
General: Removed \xpt stuff ...	211	\makebox: modified	285
1993/11/24 ltlogos.dtx LaTeX2e		\mbox: extra group	286
\LaTeX: Macro changed	80	\minipage: Redefined to support	
1993/11/28 ltclass.dtx v0.2h		extra optional arguments ...	292
\@twoclasseserror: Macro added	467	\newsavebox: Pass the whole of	
General: Assorted commands now		arg 1 to \@ifdefinable	287
in the kernel removed.	456	\parbox: Redefined to support	
Directory syntax checing moved		extra optional arguments ...	290
to dircheck.dtx	456	\raisebox: redefined to support	
Primitive filenames now		\height	294
terminated by space not		\sbox: color support	288
\relax.	456	extra group	288
\endfilecontents: Don't globally		\set@color: color support	287
allocate a write stream (always		macro added	287
use 15)	467	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. ...	466
parser from dircheck	88	General: \@onlypreamble: Many	
1993/11/29 ltoutput.dtx v1.0b		commands declared.	456
\@makecol: \@makespecialcolbox		Removed obsolete	
added	406	\@documentclass	456
\@makespecialcolbox: Command		1993/12/03 lterror.dtx v1.0b	
added	407	\@latexerr: Set	
1993/11/29 ltplain.dtx LaTeX2e		\@errorcontextlines to -1 .	60
General: All accents in decimals;		1993/12/03 ltffsini.dtx v2.1a	
suggested by Paul Taylor	29	General: update for LaTeX2e ...	208
1993/11/30 ltoutput.dtx v1.0c		1993/12/04 ltfiles.dtx v0.9b	
\fl@tracemessage: Commands		\@iinput: Macro reimplemented .	88
added	438	\@input: Macro reimplemented .	88
1993/12/01 fontdef.dtx v2.1a		\IfFileExists: Macro added ...	87
General: Update for LaTeX2e ...	214	\input: Macro reimplemented ..	88
1993/12/01 ltoutput.dtx v1.0e		\InputIfFileExists: Macro	
\@reinserts: Command added .	407	added	88
1993/12/03 ltboxes.dtx v0.1a		1993/12/05 ltfloat.dtx LaTeX2e	
\@argsbox: macro removed	294	\@dblfloatplacement: Command	
\@begin@tempboxa: macro added	286	changed	364
\@end@tempboxa: macro added ..	286	\@xfloat: Command changed ..	358
\@irsbox: redefined to support		1993/12/05 ltoutput.dtx v1.0f	
\height	295	\@addtobot: Command changed .	418

<code>\@addtocurcol</code> : Command	1993/12/07 ltclass.dtx v0.2m
changed	419
<code>\@addtodblcol</code> : Command	<code>\@fileswithoptions</code> : Reset
changed	430
<code>\@addtonextcol</code> : Command	<code>\CurrentOption</code>
changed	426
<code>\@addtotoporbot</code> : Command	1993/12/07 ltoutenc.dtx 1.1
changed	418
<code>\@boxfpsbit</code> : Command added .	441
<code>\@flcheckspace</code> : Command	General: Protected all special
added	443
<code>\@flsetnum</code> : Command added .	442
<code>\@flsettextmin</code> : Command	characters with <code>\string</code>
added	443
<code>\@flstop</code> : Commands added . . .	439
<code>\@flupdates</code> : Command added .	444
<code>\@fpsadddefault</code> : Command	1993/12/07 ltoutenc.dtx v1.1
added	440
<code>\@getfpsbit</code> : Command added .	441
<code>\@opcol</code> : Command changed . . .	405
Hook added	405
<code>\@outputpage</code> : Command	General: Made all character
changed	409
<code>\@resethfps</code> : Command added .	442
<code>\@setfloattypecounts</code> : Command	numbers decimal.
added	440
<code>\@setfpsbit</code> : Command added .	441
<code>\@shipoutsetup</code> : Command	Removed a lot of equal signs
added	409
<code>\@startcolumn</code> : Command	and the like.
changed	413
<code>\@startdblcolumn</code> : Command	1993/12/08 ltboxes.dtx v0.1b
changed	413
<code>\@testfp</code> : Command added	441
<code>\@textfloatsheight</code> : Commands	<code>\@begin@tempboxa</code> : Extra braces
added	440
<code>\@topnewpage</code> : Commands	for color support (braces
changed	398
<code>\@tryfcolumn</code> : Command	removed from other macros)
changed	414
<code>\@writesetup</code> : <code>\@startpagehook</code>	<code>\@irsbox</code> : fix typo
added	409
<code>\@output</code> : Command changed . . .	399
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>	461
1993/12/06 ltoutput.dtx v1.0f	<code>\@begin@tempboxa</code>
<code>\@specialoutput</code> : Unboxing of 255	291
added to rescue writes	399
1993/12/06 ltoutput.dtx v1.0g	<code>\lrbox</code> : move <code>\@endpefalse</code> out of
<code>\@topnewpage</code> : <code>\@floatplacement</code>	the inner group
placement bug fixed	398
1993/12/07 ltclass.dtx v0.2l	288
<code>\ProvidesFile</code> : Macro added . . .	459
	1993/12/08 ltfntcmd.dtx v2.1b
	General: Macros <code>\rm</code> , <code>\bf</code> and <code>\sf</code>
	moved to classes.dtx
	243
	1993/12/08 ltlists.dtx LaTeX2e
	<code>\@item</code> : use <code>\sbox</code> to support
	colour
	281
	1993/12/08 ltspace.dtx LaTeX2e
	<code>\@bsphack</code> : Command
	reimplemented
	71
	Command reimplemented; late
	birthday present for Chris . . .
	71
	<code>\@vbsphack</code> : Command added . . .
	73
	1993/12/09 ltboxes.dtx v0.1c
	<code>\@irsbox</code> : fix another typo
	294
	1993/12/09 ltclass.dtx v0.2n
	<code>\documentstyle</code> : input 209
	compatibility file.
	462
	1993/12/09 ltfiles.dtx v0.9e
	<code>\document</code> : Hook added
	83
	1993/12/09 ltmiscen.dtx v0.9e
	<code>\enddocument</code> : Hook added
	250
	1993/12/10 ltoutenc.dtx v1.2
	General: Added source code for
	<code>tlenc.sty</code>
	91
	1993/12/11 ltfntcmd.dtx v3.0a
	General: Complete reworking of all
	text commands, using just one
	creator function
	236
	italic correction now put in front
	of penalty before glue
	236
	newcommands replaced by defs . . .
	236
	<code>newfontswitch</code> command
	corrected and changed
	236

<code>\DeclareTextFontCommand</code> : Macro changed	238	<code>\IfFileExists</code> : Removed interactive prompting for current directory syntax	10
<code>\emph</code> : Macro changed	239	<code>\strip@prefix</code> : modified, name changed from <code>\stripmeaning</code> . . .	5
<code>\fix@penalty</code> : Macro added	241	1993/12/13 <code>ltxlists.dtx</code> latex2e	
<code>\maybe@ic</code> : Macro name changed	240	<code>\trivlist</code> : Initialised	
<code>\maybe@ic@</code> : Macro and name changed	240	<code>\@itemlabel</code>	278
<code>\sw@slant</code> : Macro changed	241	1993/12/13 <code>ltxmiscen.dtx</code> v0.9h	
<code>\textup</code> : Macros changed	239	<code>\@noligs</code> : Readded <code>\@noligs</code> . . .	257
1993/12/11 <code>ltxmath.dtx</code> v0.9g		<code>\@verbatim</code> : Readded <code>\@noligs</code> .	255
General: Added a group around the first argument of <code>\frac</code> to prevent changes (for example font changes) from modifying the contents of the second argument.	264	Removed optional argument of <code>\item</code>	255
1993/12/11 <code>ltxoutenc.dtx</code> v1.2a		<code>center</code> : Removed optional argument of <code>\item</code>	254
General: Corrected for <code>tlenc</code> , <code>math</code>	91	<code>flushleft</code> : Removed optional argument of <code>\item</code>	254
1993/12/11 <code>ltxsect.dtx</code> LaTeX2e		<code>flushright</code> : Removed optional argument of <code>\item</code>	254
<code>\@author</code> : Added default	344	1993/12/13 <code>ltxoutenc.dtx</code> v1.2b	
<code>\@title</code> : Added default	344	General: Corrected file name in driver code.	91
1993/12/11 <code>ltxref.dtx</code> LaTeX2e		1993/12/13 <code>ltxtab.dtx</code> latex2e	
<code>\@setref</code> : Macro added	246	<code>\tabbing</code> : Removed optional argument of <code>\item</code>	302
<code>\pageref</code> : Macro reimplemented .	246	1993/12/14 <code>ltxoutput.dtx</code> v1.0i	
<code>\ref</code> : Macro reimplemented	246	General: Section added to declare all parameters	449
1993/12/12 <code>ltxoutput.dtx</code> v1.0h		1993/12/15 <code>ltxboxes.dtx</code> v0.1d	
<code>\@cflb</code> : <code>boxmaxdepth</code> setting moved	412	<code>\@iminipage</code> : Changed default from ‘c’ to ‘s’	292
defs changed to <code>lets</code>	412	<code>\@iparbox</code> : Changed default from ‘c’ to ‘s’	291
<code>\@cflt</code> : name changed	412	<code>\minipage</code> : Changed default from ‘c’ to ‘s’	292
<code>\@docclearpage</code> : defs changed to <code>lets</code>	404, 405	extra space removed.	292
<code>\@makecol</code> : defs changed to <code>lets</code> .	406	<code>\parbox</code> : Changed default from ‘c’ to ‘s’	290
<code>\@resethfps</code> : Warnings added: minimal	442	1993/12/15 <code>ltxclass.dtx</code> v0.2p	
<code>\@startdblcolumn</code> : defs changed to <code>lets</code>	414	General: Removed extra ‘.’s from <code>\@@warnings</code>	456
<code>\@topnewpage</code> : braces removed . .	398	1993/12/16 <code>ltxlogos.dtx</code> LaTeX2e	
<code>\@tryfcolumn</code> : defs changed to <code>lets</code>	415	<code>\LaTeXe</code> : Extended logo by DPC	80
<code>\fl@tracemessage</code> : Commands changed	438	1993/12/16 <code>ltxmath.dtx</code> v0.9i	
1993/12/13 <code>ltxclass.dtx</code> v0.2o		<code>\@@eqnocr</code> : use <code>\refstepcounter</code> instead of shortcut	265
General: Removed setting <code>\errorcontextlines</code> (now in <code>latex.tex</code>)	456	General: use <code>\refstepcounter</code> instead of shortcut	264
<code>\documentstyle</code> : compatibility file now <code>latex209.sty</code>	462	1993/12/16 <code>ltxmiscen.dtx</code> v0.9i	
<code>\usepackage</code> : Fixed error handling	463	General: <code>\literal</code> added	257
1993/12/13 <code>ltxdirchk.dtx</code> v0.2a		1993/12/16 <code>ltxpage.dtx</code> LaTeX2e	
General: on the ‘docstrip’ pass, do not check <code>openin</code> path	10	<code>\mark</code> : Init <code>\mark</code> at begin document	382

1993/12/16 ltspace.dtx LaTeX2e			
\@bsphack: Corrected optimisation :-)	71		
1993/12/16 lttab.dtx latex2e			
\@xhline: Measure from middle of vertical rules	316		
1993/12/17 ltclass.dtx v0.2q			
\@documentclasshook: Macro added	456		
\@fileswithoptions: Add \@compatibility hook	464		
\documentstyle: Match Alan's new code.	462		
1993/12/17 ltoutenc.dtx 1.3			
General: Added this section	94		
Removed all the hackery for use in \DeclareFontEncoding, and redid everything using \DeclareTextFoo.	105, 107		
Removed the catcode hackery, since the file is only read as a package in the preamble, and removed all the messages on the screen, which just confuse users. Replaced them by the appropriate \ProvidesPackage commands. Added XXXenc.	94		
1993/12/17 ltoutenc.dtx v1.3			
General: Added			
\EncodingSpecificAccent, \EncodingSpecificAccent-edLetter and \EncodingSpecificCommand.	91		
Made Rokicki's encoding a proper encoding scheme rather than a variant of OT1.	91		
1993/12/17 ltoutput.dtx v1.0j			
\@opcol: Hook removed	405		
\@specialoutput: Page room test added	400		
\@topnewpage: check for vsize too small added	398		
Page room test added	399		
\@writesetup: —and then removed	409		
\fl@tracemessage: tracefloatvals made a document command	438		
1993/12/17 ltpage.dtx LaTeX2e			
\mark: Removed init \mark at begin document, since it doesn't work.	382		
\rightmark: Stopgap solution to mark \leftmark and \rightmark work without			
			initializing mark until the problem is solved. 381
		1993/12/18 ltoutenc.dtx 1.3b	
		General: Fixed typos with \ProvidesPackage lines. Added the \NeedsTeXFormat line. Added the last argument to \DeclareEncoding. Moved the use of the encodings to after their declaration.	94
		Replaced the missing last argument to \DeclareFontEncoding.	105, 107
		1993/12/18 ltoutenc.dtx 1.3c	
		General: Rewrote for the new syntax of \EncodingSpecific.	105, 107
		Split \EncodingSpecificAccent up into \EncodingSpecific and \DeclareAccent.	94
		1993/12/18 ltoutenc.dtx v1.3a	
		General: Replaced OT3 by XXX	91
		1993/12/18 ltoutenc.dtx v1.3b	
		General: Corrected typos.	91
		Replaced the missing last argument to \DeclareFontEncoding.	91
		1993/12/18 ltoutenc.dtx v1.3c	
		General: A new syntax, separating accent-definitions from encoding-specific definitions, and allowing encoding-specific \chardef, \let, etc.	91
		Rewrote for the new syntax of \EncodingSpecific.	91
		1993/12/18 ltoutenc.dtx v1.3d	
		General: Some T1 stuff had drifted into the OT1 file.	91
		1993/12/18 ltpage.dtx LaTeX2e	
		\sloppy: Added	
		\emergencystretch	382
		1993/12/19 ltclass.dtx v0.2r	
		\endfilecontents: Different message when ignoring a file	467
		1993/12/19 ltfntcmd.dtx v3.0b	
		General: \pdef command added	236
		Added by ASAJ.	243
		Made \@newfontswitch produce an error if command already exists, and added	
		\@renewfontswitch, ASAJ	236
		Other tidying	236
		Some more tidying done	236

Untidying added, so this is now a TEMPORARY version.	236	<code>\math@version</code> : New math font setup	147
Wording changes by CAR.	243	1994/01/17 ltfssini.dtx v2.1e <code>\not@math@alphabet</code> : Message changed	209
<code>\DeclareOldFontCommand</code> : Corrected and tidied	243	1994/01/17 ltfstrc.dtx v2.3a General: New math font setup	159
<code>\DeclareTextFontCommand</code> : Corrected and tidied	238	<code>\check@mathfonts</code> : New math font setup	168
1993/12/19 ltspace.dtx LaTeX2e <code>\@bsphack</code> : There seem to be problems with selfmade birthday presents	72	<code>\glb@currsize</code> : New math font setup	165
1993/12/20 ltdefns.dtx LaTeX2e <code>\@reargdef</code> : Kept old version of <code>\@reargdef</code> , for array.sty	39	<code>\restglb@settings</code> : New math font setup	168
1993/12/20 ltfiles.dtx v0.9m <code>\@obsoletefile</code> : Added this command, removed <code>@oldfilewarning</code>	89	1994/01/18 ltbibl.dtx LaTeX2e <code>\bibliography</code> : Use <code>\@input@</code> so include files are listed.	378
1994/01/05 fontdef.dtx v2.1d General: Removed nf prefix from file names.	216	1994/01/18 ltclass.dtx v0.2t <code>\@ifclassloaded</code> : Fix typo <code>\@pkgetension</code>	457
1994/01/13 ltmath.dtx v0.9o <code>\@eqnocr</code> : correcting 0.9i	265	1994/01/18 ltfiles.dtx v0.9p <code>\@iffileonpath</code> : Macro added	87
General: correcting 0.9i	264	<code>\@input</code> : do not use a different definition for <code>\input@path</code>	88
1994/01/14 ltdirchk.dtx v0.2d <code>\IfFileExists</code> : Close the texsys.aux output stream	10	<code>\@input@</code> : Macro added	88
1994/01/15 ltfiles.dtx v0.9o <code>\document</code> : move <code>\@preamblecmds</code> after document hook	84	<code>\IfFileExists</code> : New Definition	87
1994/01/17 ltclass.dtx v0.2s <code>\@fileswithoptions</code> : Modify to reduce parameter stack usage	464, 465	<code>\include</code> : Use <code>\@input@</code> so include files are listed.	86
General: Added many more <code>\@onlypreamble</code> commands	456	<code>\InputIfFileExists</code> : New Definition	88
Wrapped long lines to column 72	456	1994/01/18 ltfssini.dtx v2.1f <code>\not@math@alphabet</code> : Message corrected	209
1994/01/17 ltfiles.dtx LaTeX2e <code>\listfiles</code> : New Version, adds ' <code>.tex</code> ' if needed, and lines up columns	89	1994/01/18 ltmiscen.dtx v0.9p <code>\@verbatim</code> : Add <code>\global\@inlabelfalse</code>	255
1994/01/17 ltfssbas.dtx v2.1a General: New math font setup	139	Only add <code>\penalty</code> if in hmode	255
<code>\curr@math@size</code> : New math font setup	148	1994/01/19 fontdef.dtx v2.1e General: Added missing setting for symbols in bold version.	219
<code>\everydisplay</code> : New math font setup	148	1994/01/19 ltdirchk.dtx v0.2e <code>\IfFileExists</code> : name changed from <code>\test</code>	9
<code>\everymath</code> : New math font setup	148	<code>\input@path</code> : No longer check that an empty group is in the path	11
<code>\frozen@everydisplay</code> : New math font setup	148	<code>\strip@prefix</code> : name changed from <code>\strip@meaning</code> , to match NFSS.	5
<code>\frozen@everymath</code> : New math font setup	148	1994/01/19 ltmath.dtx v1.0n classes <code>\mathindent</code> : Deferred setting of <code>\mathindent</code>	267
		1994/01/20 ltdirchk.dtx v0.2f General: <code>\@copytexsys</code> and the texsys.new file removed	9
		Modify all of ltxcheck	13

\IfFileExists: \@copytexsys removed	10	1994/01/31 ltfntcmd.dtx v3.1b	
1994/01/21 ltclass.dtx v0.2u		General: \@normalsize no longer defined	236
\documentstyle: compatibility file now latex209.def.	462	1994/02/01 ltpage.dtx LaTeX2e	
1994/01/21 ltdirchk.dtx v0.2g		\pagestyle: (DPC) Modify to get nicer error message	380
General: Improve documentation, reorganise docstrip module . . .	1	\thispagestyle: (DPC) Modify to get nicer error message	381
\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.	465
1994/01/21 ltfloat.dtx LaTeX2e		1994/02/03 ltoutput.dtx v1.0k	
\@xfloat: Added missing percent characters.	358	\@makespecialcolbox: correct mistakes in the documentation	408
1994/01/21 ltmiscen.dtx v0.9s		1994/02/07 ltclass.dtx v0.2y	
\verbatim@font: Removed unnecessary category code hackery.	256	\@fileswithoptions: Run \@compatibility on the first class to start (not the first to finish)	464
1994/01/24 ltdirchk.dtx v0.2h		\@ifclasswith: Add extra ,s so ‘two’ is not matched with ‘twocolumn’	458
\IfFileExists: Stop testing once texsys.aux has been found	9	\@ProcessOptions*: Add extra ,s so ‘two’ is not matched with ‘twocolumn’	460, 461
1994/01/24 ltpage.dtx LaTeX2e		1994/02/07 ltfssbas.dtx v2.1c	
\pagestyle: (DPC) Complain if pagestyle is undefined.	380	\DeclareFontEncoding: revert catcode settings earlier	141
1994/01/25 ltdirchk.dtx v0.2i		\DeclareFontShape: revert catcode settings earlier	140
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	408
\math@version: Corrections for math setup	148	boxmaxdepth setting removed	407
1994/01/25 ltmath.dtx LaTeX2e		General: Documentation and tasks tidied.	383
\bordermatrix: Removed		1994/02/10 ltclass.dtx v0.2z	
\p@renwd.	261	\@documentclasshook: Changed the name from \@compatibility to \@documentclasshook, and added the check for whether \@normalsize has been defined. ASAJ.	456
1994/01/26 ltfssstrc.dtx v2.3c		\@fileswithoptions: Renamed \@compatibility to \@documentclasshook. ASAJ.	464
\check@mathfonts: Correct trace info placement	168	1994/02/10 ltfssbas.dtx v2.1d	
\restglb@settings: Correct trace info placement	168	\addto@hook: Made \addto@hook long.	157
1994/01/27 ltfntcmd.dtx v3.1a			
\nocorrlist: Only ., used as default for cm fonts	242		
1994/01/29 ltclass.dtx v0.2v			
\@unprocessedoptions: Macro added.	467		
\@fileswithoptions: All options raise error if no \@ProcessOptions appears . . .	465		
1994/01/31 ltclass.dtx v0.2w			
\g@addto@macro: Use toks register to avoid ‘hash’ problems . . .	466		
1994/01/31 ltfiles.dtx v0.9t			
\document: set \@normalsize or \@normalsize if necessary	84		

1994/02/10 ltffscmp.dtx v2.1d	Long lines wrapped to 72 columns	81
\scan@fontshape: scan away stuff after pt		182
1994/02/22 ltffsini.dtx v2.1g	General: Correct error message . .	211
1994/02/24 ltffsbas.dtx v2.1e	\DeclareFontShape: Separate restoration of catcodes for fd cmds	140
	\define@newfont: Separate restoration of catcodes for fd cmds	150
	\nfss@catcodes: Separate restoration of catcodes for fd cmds	150
1994/02/25 ltdirchk.dtx v0.2j	General: Remove need for drv file .	1
1994/03/01 ltdirchk.dtx v0.2k	General: Add unstripped module, so that dircheck.dtx may be used with initex	1
1994/03/02 ltboxes.dtx v0.1e	General: Add 2kernel module . .	285
	Remove need for drv file	285
1994/03/02 ltclass.dtx v0.3a	General: Remove need for driver file	456
1994/03/03 ltboxes.dtx v0.1f	\@irsbox: Replaced a missing \else	294
1994/03/04 ltfloat.dtx v1.0a	General: Initial version, split from latex.dtx	354
1994/03/04 ltsect.dtx v1.0a	General: Initial version, split from latex.dtx	344
1994/03/04 lttab.dtx v1.0a	General: Initial version, split from latex.dtx	296
1994/03/04 ltvers.dtx v1.0a	General: Initial version, split from latex.dtx	32
1994/03/07 ltboxes.dtx v0.1a	\@mpfootnotetext: Extra group for colour	293
1994/03/07 ltboxes.dtx v1.0a	General: Unify format with other Kernel files	285
1994/03/07 ltdefs.dtx v1.0a	\@@italiccorr: Macro added . . .	35
1994/03/07 ltfiles.dtx v1.0a	General: Initial version, split from latex.dtx	81
1994/03/07 ltfinal.dtx v0.1a	General: Add code from the old dump.dtx	504
	Initial version, split from latex.dtx	496
	move code here from lhyphen.dtx	501
	Remove oldcomments environment	496
	use \InputIfFileExists not \IfFileExists	501
1994/03/07 ltfloat.dtx v1.0a	\@endfloatbox: (DPC) Extra group for colour	363
	\@footnotetext: (DPC) Extra group for colour	372
	\@xfloat: (DPC) Extra group for colour	359
1994/03/07 lthyphen.dtx v0.1c	General: move the 2kernel code to ltfinal.dtx	471
1994/03/07 ltlength.dtx v1.0a	\@settodim: (DPC) Extra group for colour	138
1994/03/07 ltlists.dtx v1.0a	General: Initial version, split from latex.dtx	270
	Long lines wrapped to 72 columns	270
1994/03/07 ltpage.dtx v1.0a	General: Initial version, split from ltherest.dtx	380
1994/03/07 ltpictur.dtx v0.1a	General: Initial version, split from latex.dtx	318
	Long lines wrapped to 72 columns	318
1994/03/07 ltsect.dtx v1.0a	\@changfrom: (DPC)Extra groups for colour	350
1994/03/07 lttab.dtx v1.0a	General: Long lines wrapped to 72 columns	296
1994/03/08 ltclass.dtx v0.3b	General: Modify driver code into ‘new style’	456
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.	14

1994/03/10 ltssbas.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 88
\backslash bgroup/ \backslash egroup. 156	1994/03/13 ltssbas.dtx v2.1g
1994/03/11 ltssdcl.dtx v2.1b	General: add 2ekernel module to
\backslash DeclareSymbolFontAlphabet@:	omit repeated code 139
Added check against use of	1994/03/13 ltssdcl.dtx v2.1c
alphabet switch outside of	General: add 2ekernel module to
math mode. 207	omit repeated code 185
\backslash SetMathAlphabet@: Changed	1994/03/14 ltboxes.dtx v1.0b
parameter template in	\backslash @isavebox: Use
temporary macro to catch	\backslash color@setgroup 288
check add below. 198	\backslash @isavepicbox: Use
1994/03/12 ltclass.dtx v0.3c	\backslash color@setgroup 288
\backslash @fileswithoptions: Do not use	\backslash color@begingroup: macro added
\backslash @pr@videpackage to avoid	for colour support 287
timeout 465	\backslash color@endgroup: macro added for
General: Change name from	colour support 287
docclass to ltclass 456	\backslash lrbox: Use \backslash color@setgroup . . 288
\backslash ProvidesFile: Add \backslash wlog 459	\backslash sbox: Use \backslash color@setgroup . . . 288
\backslash ProvidesPackage: Add \backslash wlog . . 458	1994/03/14 ltfloat.dtx 1.0c
use \backslash @gtempa 458	\backslash @xympar: (DPC) Use
1994/03/12 ltdefns.dtx v1.0b	\backslash color@begingroup 367
\backslash @reargdef: New defn, in terms of	1994/03/14 ltfloat.dtx v1.0c
\backslash @yargdef 39	\backslash @endfloatbox: (DPC) Use
\backslash @yargd@f: Name changed from	\backslash color@endgroup 363
\backslash XXX@argdef 38	\backslash @footnotetext: (DPC) Use
1994/03/12 ltdirchk.dtx v1.0b	\backslash color@begingroup, add
General: Change name from	\backslash endgraf 372
dircheck.dtx 1	\backslash @savemarbox: (DPC) Use
Minor edits to the timeouts in	\backslash color@begingroup 366
ltxcheck 1	\backslash @xfloat: (DPC) Use
1994/03/12 ltfloat.dtx v1.0b	\backslash color@begingroup 359
\backslash @savemarbox: (DPC) Extra group	1994/03/15 ltfiles.dtx LaTeX2e
for colour 366	\backslash @missingfileerror: Quit on x or
\backslash @xympar: (DPC) Extra bgroup for	X just like a real error 88
colour 367	1994/03/15 ltfntcmd.dtx v3.2a
1994/03/12 ltplain.dtx v1.0b	General: Adapted to mass
General: Name changed from	formatting 236
lplain. The end of an era 14	Changed \backslash / to \backslash @@italiccorr 236
1994/03/12 ltplain.dtx v1.0e	Removed \backslash @renewfontswitch . 236
General: Replaced remaining	Removed defs of short-forms and
width, height, depth by L ^A T _E X	all sizes except \backslash normalize . 236
macro names to save tokens. . . 14	1994/03/15 ltoutput.dtx v1.0l
1994/03/13 ltcntrl.dtx v1.0c	\backslash @addtocurcol: Changed
\backslash @tfor: (DPC) Add \backslash @t@r so a	\backslash addvspace to \backslash vskip . . 421, 425
single group is correctly	\backslash @combinedblfloats: Removed
treated. 54	boxmaxdepth setting. 413
1994/03/13 ltfiles.dtx LaTeX2e	\backslash @makecol: \backslash maxdepth changed to
\backslash @addtofilelist: Macro added . . 89	\backslash @maxdepth 406
\backslash listfiles: Reset	Removed boxmaxdepth setting. 407
\backslash @addtofilelist at begin	\backslash @makespecialcolbox: Removed
document 89	boxmaxdepth setting. 408

<code>\@topnewpage</code> : Corrected and amended warning message . . .	398	1994/03/28 ltsect.dtx v1.0b	General: Split further from ltherest.dtx	344
Warning added: it should be improved	399	1994/03/28 lttab.dtx v1.0b	General: Improve documentation	296
General: Added some warnings when page gets full of top floats.	383	1994/03/28 ltthm.dtx v1.0a	General: Initial version, split from latex.dtx	340
Driver added and further tidying.	383	1994/03/29 ltcounts.dtx v1.0c	General: Create file from parts of ltmiscen and ltherest.	132
Removed duplicated code and corrected docstrip options. . .	383	1994/03/29 ltlength.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	138
Some boxmaxdepth settings removed.	383	1994/03/29 ltmiscen.dtx v1.0d	General: Remove counter macros to ltcntlen	249
1994/03/16 ltclass.dtx v0.3f		1994/03/29 ltpageno.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	244
General: Add pkgindoc package .	469	1994/03/29 ltxref.dtx v1.0c	General: Create file ltcntlen from parts of ltmiscen and ltherest.	245
1994/03/16 ltfiles.dtx LaTeX2e		1994/03/31 ltbibl.dtx v1.0a	General: Initial version of ltidxbib.dtx, split from ltherest.dtx	376
<code>\listfiles</code> : Move this code directly into <code>\document</code>	89	1994/03/31 ltidxglo.dtx v1.0a	General: Initial version of ltidxbib.dtx, split from ltherest.dtx	374
1994/03/16 ltfiles.dtx v1.0c		1994/04/09 ltcounts.dtx v1.0d	<code>\@newctr</code> : <code>\@nocnterr</code> now has counter name argument	133
<code>\document</code> : (DPC) directly add file list settings	84	<code>\addtocounter</code> : <code>\@nocnterr</code> now has counter name argument .	133	
1994/03/16 ltmiscen.dtx v1.0b		<code>\setcounter</code> : <code>\@nocnterr</code> now has counter name argument	133	
<code>\@verbatim</code> : Remove <code>\global\@inlabelfalse</code> again.	255	<code>\stepcounter</code> : Use <code>\addtocounter</code> to have name checked	133	
1994/03/28 ltalloc.dtx v1.0d		1994/04/09 ltthm.dtx v1.0b	<code>\@othm</code> : Use standard counter error message (FMi)	342
General: Redefinition of ‘new’ allocations removed.	49	1994/04/11 ltclass.dtx v0.3g	<code>\endfilecontents</code> : Add star form, dont write <code>\endinput</code> at the end of the file.	467
1994/03/28 ldirchk.dtx v1.0d		<code>\ProvidesFile</code> : Protect against weird catcodes.	459	
General: Improve documentation .	1	1994/04/11 ltfssbas.dtx v2.1h	General: Added <code>\defaultscriptratio</code> and <code>\defaultscriptscriptratio</code> . ASAJ.	139
1994/03/28 lterror.dtx v1.0d				
<code>\@invalidchar</code> : (DPC) Comment out (use <code>catcode15</code> instead) . .	62			
General: Remove test for <code>\inputlineno</code> undefined.	59			
1994/03/28 ltfiles.dtx v1.0d				
<code>\document</code> : (DPC) Use <code>\normalsize</code> not <code>\@normalsize</code>	84			
(DPC) remove <code>\@normalsize</code> check	84			
1994/03/28 ltfloat.dtx v1.0b				
<code>\@caption</code> : Use <code>\normalsize</code> not <code>\@normalsize</code>	357			
General: Split further from ltherest.dtx	354			
1994/03/28 ltlists.dtx v1.0b				
General: Improve documentation	269			
1994/03/28 ltmiscen.dtx v1.0c				
General: Improve Documentation	249			
1994/03/28 ltplain.dtx v1.0c				
<code>\newlanguage</code> : Remove some <code>\outer</code> declarations.	16			

<code>\defaultscriptratio</code> : Macro added	157	<code>\no@alphabet@error</code> : Use std LaTeX error macro	139
<code>\defaultscriptscipratio</code> : Macro added	157	1994/04/18 <code>ltfssdcl.dtx</code> ???	
1994/04/12 <code>ltboxes.dtx</code> v1.0c		<code>\DeclareMathAlphabet</code> : Pass correct arg (2 not 3)	196
General: Remove <code>\@acci</code> , now defined in <code>lplain.dtx</code>	291	1994/04/18 <code>ltfssdcl.dtx</code> v2.1d	
Remove <code>\@dischyph</code> , now defined in <code>linit.dtx</code>	291	General: Removed surplus <code>\no@alphabet@error</code> (see <code>fam.dtx</code>)	185
1994/04/12 <code>ltdefns.dtx</code> v1.0g		1994/04/18 <code>ltsfstrc.dtx</code> v2.3d	
<code>\@dischyph</code> : Define <code>\@dischyph</code> , was previously in <code>ltboxes.dtx</code>	35	General: Changed to new error/warning scheme	159
1994/04/12 <code>lplain.dtx</code> v1.0d		<code>\font@submax</code> : Changed dimen to macro	176
General: Define <code>\@acci</code>	29	<code>\fontsubfuzz</code> : Changed dimen to macro	176
1994/04/12 <code>ltvers.dtx</code> v1.0b		<code>\subst@size</code> : <code>\font@submax</code> and <code>\fontsubfuzz</code> now macros	177
General: Have version info generated automatically.	32	1994/04/19 <code>ltpage.dtx</code> v1.0b	
1994/04/14 <code>lftncmd.dtx</code> v3.2b		General: Improve documentation	380
General: Macros renamed to non-private forms, JB	236	1994/04/20 <code>lftncmd.dtx</code> v3.3a	
<code>\DeclareOldFontCommand</code> : Renamed from <code>\@newfontswitch</code>	242	General: Documentation up-dated	236
1994/04/15 <code>ltboxes.dtx</code> v1.0d		New implementation of <code>\nocorr</code>	236
<code>\@isavebox</code> : Added missing percent character.	288	<code>\check@nocorr@</code> : Macros added	239
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	241
<code>\@newctr</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	133	1994/04/20 <code>ltmiscen.dtx</code> v1.0e	
<code>\addtocounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	133	<code>\enddocument</code> : Changed logic for producing warning messages	251
<code>\setcounter</code> : Use <code>\@nocounterr</code> instead of <code>\@nocnterr</code>	133	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	292
<code>\@nocounterr</code> : New name for error message, old error message (without arg) kept	60	<code>\@mpfootnotetext</code> : Extra <code>\endgraf</code> for colour	293
1994/04/17 <code>ltthm.dtx</code> v1.0c		<code>\endminipage</code> : Extra <code>\egroup</code> for colour	293
<code>\@othm</code> : Use new std counter error message (FMi)	342	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.	496
General: Initialise <code>\textheight</code> , <code>\textwidth</code> and page style	498	1994/04/22 <code>ltfssini.dtx</code> v2.1g	
1994/04/18 <code>lfloat.dtx</code> v1.0d		<code>\not@math@alphabet</code> : Message changed again	209
<code>\@footnotetext</code> : (DPC) Remove Colour support	372	1994/04/23 <code>ltfinal.dtx</code> v0.1d	
<code>\@savemarbox</code> : (DPC) Remove Colour support	366	General: Check that <code>\font@submax</code> is still zero	496
1994/04/18 <code>ltfssbas.dtx</code> v2.1i		1994/04/24 <code>ltoutput.dtx</code> v1.0m	
General: Macro <code>\no@alphabet@help</code> removed again	139	<code>\@resetfps</code> : Number 2 changed to <code>\tw@</code>	442
<code>\calculate@math@sizes</code> : Changed message to log only	157	Warning changed	442

<code>\@specialoutput</code> : Message changed to give more info and ‘top’ removed	400	1994/04/28 ltplain.dtx v1.0g General: Turn off overfull box tracing in log	24
<code>\@topnewpage</code> : Message changed to give more info	399	1994/04/29 ltclass.dtx v1.0a General: Change version number to 1 (no other change)	456
Warning message removed as it will be generated later	398	1994/04/29 ltmiscen.dtx v1.0f <code>\@verbatim</code> : <code>\leavevmode</code> added	255
General: Changed <code>\@normalsize</code> to <code>\normalsize</code>	383	Change to <code>\everypar</code> added	255
Corrected unverbbed commands in documentation.	383	1994/04/29 ltoutenc.dtx 1.4a General: Removed <code>\EncodingSpecific</code> . Renamed all the commands. Added <code>\DeclareTextGlyph</code> and <code>\UndeclareTextCommand</code>	94
Removed some long lines and other aesthetic changes.	383	Removed Rokicki’s OT1 variant encoding. Moved the driver to the top.	94
Warning messages changed/corrected.	383	1994/04/30 ltfntcmd.dtx v3.3b General: Documentation up-dated and tidied	236
1994/04/24 ltpictur.dtx v0.1b General: Removed surplus spaces after <code>\hbox to</code> in several cases	318	Prefix <code>frag@</code> changed to <code>frag</code> in <code>\@protecteddef</code>	236
1994/04/25 ltclass.dtx v0.3h General: Removed spurious extra ‘.’s at the end of error messages	456	Title changed	236
1994/04/25 ltfloat.dtx v1.0e <code>\@largefloatcheck</code> : Changed warning message to give more info	363	Warning changed to info message in <code>\@protecteddef</code>	236
Command added	363	1994/04/30 ltoutput.dtx v1.0n <code>\@activechar@info</code> : <code>\@activechar@warning</code> changed to <code>\@activechar@info</code>	409
General: Changed warning messages	354	<code>\@combinedblfloats</code> : Removed rule in <code>topnewpage</code> case	413
Removed obsolete tracing code	354	<code>\@emptycol</code> : Empty column action added: <code>\@emptycol</code>	398
1994/04/27 ltfssstrc.dtx v2.3e General: Corrected item that was forgotten in last change.	159	<code>\@flsetnum</code> : Rogue space removed	442
1994/04/28 ltterror.dtx v1.0j <code>\@inmatherr</code> : Macro added	62	<code>\@specialoutput</code> : Cut-off point changed to <code>2\baselineskip</code>	400
1994/04/28 ltterror.dtx v1.1c <code>\@inmatherr</code> : Replaced <code>\noexpand</code> with <code>\protect</code>	62	Empty column action added: <code>\@emptycol</code>	400
1994/04/28 ltfssdcl.dtx v2.1e General: Removed all <code>\uppercase</code> in hex num parsing macros	185	Extra empty column added for <code>twocolumn</code> case	400
1994/04/28 ltlists.dtx v1.0c General: Replaced <code>\@ltxnomath</code> by <code>\@inmatherr</code>	279	Extra empty column added for <code>twocolumn</code> case (wrong, see below)	400
1994/04/28 ltpictur.dtx v0.1c General: bezier curves added	337	<code>\@topnewpage</code> : Added setting of <code>\col@number</code>	398
<code>\multiput</code> : (DPC) Ignore spaces between <code>)</code> (.	320	Cut-off point changed to <code>3\baselineskip</code>	399
(DPC) Macro added	320	Empty column action added: <code>\@emptycol</code>	399
<code>\picture</code> : (DPC) Ignore spaces before <code>(</code>	319	Message changed for Frank	399

General: <code>\@activechar@warning</code> changed to an info message.	383	1994/05/02 ltmiscen.dtx v1.0g General: Changed 91 to 1991 and moved some bits	249
Added <code>\col@number</code>	383	1994/05/02 ltoutput.dtx v1.0o <code>\@resethfps</code> : Code shortened	442
Documentation tidied.	383	General: Code of <code>\@resethfps</code> shortened.	383
Empty column action added.	383	1994/05/03 ltbibl.dtx v1.0b <code>\nocite</code> : Make <code>\nocite</code> issue a warning for an undefined citation key.	378
Fixed bug from <code>\dblfigrule</code> with <code>\@topnewpage</code>	383	1994/05/03 ltfinal.dtx v0.1f General: Set the catcode of control-J to be ‘other’, for use in messages.	496
Full of floats action improved.	383	1994/05/03 ltfloat.dtx v1.0f General: (CAR) Added <code>\@largefloatcheck</code>	354
<code>\col@number</code> : Added <code>\col@number</code>	396	Removed unnecessary braces from arguments of <code>\@ifnextchar</code>	354
<code>\onecolumn</code> : Added setting of <code>\col@number</code>	397	<code>\end@dblfloat</code> : <code>\@largefloatcheck</code> added	362
1994/05/01 lterror.dtx v1.0k <code>\@latexerr</code> : (CAR) Added draft <code>\@latexinfo</code>	60	<code>\end@float</code> : (CAR) Added <code>\@largefloatcheck</code>	361
1994/05/01 ltoutenc.dtx 1.4a General: Added the <code>\a</code> command. Added the <code>\SaveAtCatcode</code> and <code>\RestoreAtCatcode</code> commands.	102 105	1994/05/03 ltfsdcl.dtx v2.1f General: Renamed <code>\@@DeclareMathDelimiter</code> to <code>\DeclareMathDelimiter</code>	185
Removed the <code>uc/lc</code> table settings, since the T1 <code>uc/lc</code> table is now the default.	112	1994/05/03 ltlists.dtx v1.0d <code>\@item</code> : <code>\hskip</code> changed to <code>\kern</code> General: Removed superfluous braces	280 279
Rewrote for the new syntax.	105, 107	1994/05/03 ltmiscen.dtx v1.0h <code>\@centercr</code> : <code>\@badcrerr</code> replaced by <code>\@nolnerr</code>	254
1994/05/01 ltoutenc.dtx v1.4a General: Removed Rokicki’s encoding.	91	1994/05/03 lttab.dtx v1.0d <code>\@endpbox</code> : Use <code>\@finalstrut</code> based on depth of <code>\@arstrutbox</code>	317
Renamed the commands, removed the <code>\EncodingSpecific</code> command. Turned all slots into decimal. Added <code>\a</code>	91	1994/05/04 ltclass.dtx v1.0b <code>\NeedsTeXFormat</code> : Changed wording of the warning	463
1994/05/02 ltcntrl.dtx v1.0l <code>\@break@tfor</code> : Macro added (from ltfiles.dtx)	54	1994/05/04 lterror.dtx v1.0m <code>\@badcrerr</code> : Error message removed	62
1994/05/02 ltfiles.dtx v1.0f <code>\@iffilenamepath</code> : <code>\@break@loop</code> renamed to <code>\@break@tfor</code>	87	1994/05/05 ltbibl.dtx v1.0c <code>\@citex</code> : Set switch for warning and end of run.	377
<code>\@obsoletefile</code> : Make <code>\@onlypreamble</code>	89	<code>\nocite</code> : Do not write page number in <code>\nocite</code> warning message.	378
1994/05/02 ltfinal.dtx v0.1e General: Added setting the ‘letter’ catcodes.	502	Set switch for warning and end of run.	378
Added setting the ‘other’ catcodes.	502		
Added setting the special catcodes.	502		
Made slot 127 illegal	502		
Set all the catcodes	496		
1994/05/02 ltfinal.dtx v0.1f General: Set the catcode of control-J.	502		

- 1994/05/05 ltfinal.dtx v0.1g
 General: Added empty errhelp. . . 496
 \backslash errhelp: Set error help empty. . . 506
- 1994/05/05 ltfntcmd.dtx v3.3c
 \backslash @math@egroup: Corrected
 \backslash @fontswitch and added saved
 versions 243
 General: Corrected \backslash @fontswitch 236
- 1994/05/05 ltmiscen.dtx v1.0i
 General: Removed braces from
 ifnextchar and ifstar
 arguments 249
- 1994/05/07 lttab.dtx v1.0c
 \backslash @maxtab: Changed \backslash @firsttab to
 \backslash chardef 300
 Changed \backslash @maxtab to \backslash chardef 300
 General: Removed definition of \backslash + 296
 Removed surplus braces from
 \backslash @ifnextchar constructs . . . 296
- 1994/05/08 ltfntcmd.dtx v3.3d
 General: Removed
 \backslash @undefinedfonterror 236
 \backslash normalsize: Removed
 \backslash @undefinedfonterror 243
- 1994/05/09 ltfntcmd.dtx v3.3f
 General: Replaced all \backslash next by
 \backslash @let@token and undo change
 3.3e, whatever that was. . . . 236
- 1994/05/10 ltdefs.dtx v1.0n
 General: (ASAJ) Added
 \backslash DeclareProtectedCommand. . . 34
 Added
 \backslash DeclareProtectedCommand . . . 42
 Added \backslash makeatletter and
 \backslash makeatother ASAJ. 47
 Removed braces around
 \backslash @ifundefined argument.
 ASAJ. 39
- 1994/05/10 lterror.dtx v1.0n
 \backslash @latexerr: (ASAJ) Added extra
 blank lines to \backslash @latexerr. . . . 60
- 1994/05/10 ltmiscen.dtx v1.0j
 \backslash @sverb: Slight change in error
 message text. 256
- 1994/05/11 ltboxes.dtx v1.0f
 \backslash @begin@tempboxa: Use new
 \backslash color@setgroup concept. . . . 286
 \backslash @iiiminipage: Use new
 \backslash color@setgroup concept. . . . 292
 \backslash @mpfootnotetext: Use new
 \backslash color@setgroup concept. . . . 293
 Use new \backslash normalcolor and
 \backslash @finalstrut. 293
- General: Superfluous braces
 removed from several
 commands 285
- \backslash color@setgroup: macro added for
 colour support 287
- \backslash endminipage: Use new
 \backslash color@setgroup concept. . . . 293
- 1994/05/11 ltclass.dtx v1.0c
 \backslash endfilecontents: Add checks for
 form feed and tab 467
- 1994/05/11 ltdirchk.dtx v1.0e
 General: Add \backslash ProvidesFile as
 used in fd files. 4
- 1994/05/11 lterror.dtx v1.0o
 \backslash @latexerr: (ASAJ) Removed one
 of the extra blank lines to
 \backslash @latexerr. 60
- 1994/05/11 ltlogos.dtx v1.0o
 \backslash LaTeX: Use
 \backslash DeclareProtectedCommand.
 ASAJ. 80
 \backslash LaTeXe: Use
 \backslash DeclareProtectedCommand.
 ASAJ. 80
- 1994/05/11 ltoutenc.dtx 1.5a
 General: Made T1 and OT1
 generate packages rather than
 def files. Renamed the
 ‘package’ module to ‘teststy’. . . 94
- 1994/05/11 ltoutenc.dtx v1.5a
 General: Reimplemented
 \backslash DeclareTextCommand using
 \backslash @changed@cmd and
 \backslash DeclareProtectedCommand. . . 94
 Renamed the commands again.
 Made the encoding part of the
 command syntax. Added the
 \backslash DeclareTextCommand
 interface. Used
 \backslash DeclareProtectedCommand. . . 91
 \backslash DeclareTextAccent:
 Reimplemented using
 \backslash DeclareTextCommand. 97
- 1994/05/11 ltspaced.dtx v1.0o
 \backslash ,: Use \backslash DeclareRobustCommand.
 ASAJ. 78
 \backslash hspace: Use
 \backslash DeclareRobustCommand.
 ASAJ. 79
- 1994/05/12 ltboxes.dtx v1.0g
 \backslash @finalstrut: macro added 295
 \backslash fbox: New definition, merged
 with \backslash framebox 289

<code>\framebox</code> : Merged <code>\fbox</code> and <code>\framebox</code>	289	<code>\RestoreAtCatcode</code> commands.	105
<code>\normalcolor</code> : macro added for colour support	287	Rewrote for the new syntax.	105, 107
1994/05/12 ltdefns.dtx v1.0p General: (ASAJ) Fixed a bug with <code>\relax</code> which was using <code>\@gobble</code> before defining it. . .	34	1994/05/12 ltoutput.dtx v1.0p <code>\@writsetup</code> : <code>\normalcoloradded</code>	409
Fixed a bug with <code>\relax</code> which was using <code>\@gobble</code> before defining it.	42	General: <code>\normalcoloradded</code> in various places (DPC).	383
1994/05/12 ltfsbas.dtx v2.1j General: New <code>baselinestretch</code> concept	139	1994/05/13 ltboxes.dtx v1.0h <code>\@arrayparboxrestore</code> : New accent system, use <code>\let</code> not <code>\def</code>	292
Replaced hand-protected commands by <code>\DeclareRobustCommand</code> defs	139	1994/05/13 ltcounts.dtx v1.0f General: Removed <code>\@Ialph</code>	134
<code>\f@linespread</code> : New macro	147	Removed <code>\@ialph</code>	134
<code>\fontencoding</code> : Use <code>\DeclareRobustCommand</code>	145	1994/05/13 ltdefns.dtx v1.0q General: (ASAJ) Renamed <code>\DeclareProtectedCommand</code> to <code>\DeclareRobustCommand</code> . Removed <code>\@if@short@command</code>	34
<code>\fontfamily</code> : Use <code>\DeclareRobustCommand</code>	146	(ASAJ) Replaces <code>\space</code> by ‘ ’ in <code>\csname</code>	34
<code>\fontseries</code> : Use <code>\DeclareRobustCommand</code>	146	Renamed <code>\DeclareProtectedCommand</code> to <code>\DeclareRobustCommand</code> . Removed <code>\@if@short@command</code> . Moved to after the definition of <code>\@gobble</code>	42
<code>\fontshape</code> : Use <code>\DeclareRobustCommand</code>	146	1994/05/13 ltdefns.dtx v1.0r General: (ASAJ) Added logging message to <code>\DeclareProtectedCommand</code> . . .	34
<code>\fontsize</code> : Redefined to use <code>\set@fontsize</code>	147	Added logging message to <code>\DeclareProtectedCommand</code> . . .	42
<code>\linespread</code> : New macro	147	1994/05/13 ltdefns.dtx v1.0s General: (ASAJ) Added <code>\@backslashchar</code>	34
<code>\mathversion</code> : Use <code>\DeclareRobustCommand</code>	147	(ASAJ) Coded <code>\@ifdefinable</code> more efficiently.	34
1994/05/12 ltfsdcl.dtx v2.1g General: Allow <code>\relax</code> as undefined command	185	Coded more efficiently, thanks to FMi.	39
Allow <code>\relax</code> 'ed cmds to be declared	185	1994/05/13 ltfiles.dtx LaTeX2e <code>\listfiles</code> : Stop <code>\listfiles</code> being run twice	89
1994/05/12 ltfsini.dtx v2.1i General: Moved <code>\fontencoding</code> to fam.dtx	208	1994/05/13 ltfiles.dtx v1.0g <code>\document</code> : Added execution of <code>\every@size</code>	84
Moved <code>\fontfamily</code> to fam.dtx	208	1994/05/13 ltfinal.dtx v0.1h General: Added package <code>otlenc</code> , and defined <code>\@acci</code> , <code>\@accii</code> and <code>\@acciii</code>	496
Moved <code>\fontseries</code> to fam.dtx	208		
Moved <code>\fontshape</code> to fam.dtx	208		
Moved <code>\fontsize</code> to fam.dtx .	208		
Moved <code>\mathversion</code> to fam.dtx	208		
Moved <code>\selectfont</code> to tracefnt.dtx	208		
1994/05/12 ltfsstrc.dtx v2.3f <code>\selectfont</code> : Use <code>\DeclareRobustCommand</code>	163		
1994/05/12 ltoutenc.dtx 1.5a General: Removed the <code>\SaveAtCatcode</code> and			

1994/05/13 ltfinal.dtx v1.0h		<code>\DeclareFontEncoding</code> : Log if encoding is redeclared	142
General: Added output enc stuff .	504	Only init enc change cmd when new encoding	142
1994/05/13 ltfloat.dtx v1.0g		1994/05/14 ltfsini.dtx v2.1k	
<code>\@footnotetext</code> : (DPC) Add new style colour support:		General: Init error font just before checking for fontdef.cfg	212
<code>\normalcolor</code>	372	<code>\p@reset@font</code> : Remove surplus braces	211
(DPC) Use <code>\@finalstrut</code>	372	1994/05/14 ltfsstrc.dtx v2.3h	
<code>\@xfloat</code> : (DPC) Use		<code>\selectfont</code> : Added	
<code>\normalcolor</code>	359	<code>\enc@update</code>	164
1994/05/13 ltfntcmd.dtx v3.3g		1994/05/14 ltoutenc.dtx 1.5d	
General: Replaced <code>\@protecteddef</code> by <code>\DeclareRobustCommand</code> .	236	General: Moved the driver to the top.	94
1994/05/13 ltfsbas.dtx v2.1k		1994/05/14 ltoutenc.dtx v1.5c	
General: Remove File identification ‘typeout’	139	General: Added the fontenc package	119
1994/05/13 ltfsbas.dtx v2.1l		Added the fontenc package.	91
<code>\DeclareFontEncoding</code> : Init encoding change command .	142	Fixed a bug which caused an infinite loop if <code>\f@encoding</code> was incorrectly set.	91, 94
<code>\define@newfont</code> : Use <code>\@input@</code> for fd files	150	Moved fontsmpl to its own dtx file.	91
1994/05/13 ltfsdcl.dtx v2.1h		1994/05/14 ltoutenc.dtx v1.5d	
General: Removed file identification typeout	185	General: Rewrote	
1994/05/13 ltfsini.dtx v2.1j		<code>\DeclareTextCommand</code> to define its argument to use the current encoding by default, rather than the encoding provided to <code>\DeclareTextCommand</code>	91, 94
General: Removed file identification typeout	208	Tidied up the documentation.	91
1994/05/13 ltfsstrc.dtx v2.3g		1994/05/14 ltoutenc.dtx v1.5e	
General: Removed typeouts as <code>\ProvidesPackage</code> writes to log.	159	General: Replaced <code>\ENC@cmd</code> by <code>\ENC-cmd</code>	91
1994/05/13 ltoutenc.dtx v1.5b		1994/05/15 ltfsbas.dtx v2.1o	
General: Added <code>\{, \}</code> and <code>\\$</code>	91	General: encoding cmds changed to enc-cmd	139
Renamed		1994/05/16 ltalloc.dtx v1.1a	
<code>\DeclareProtectedCommand</code> to <code>\DeclareRobustCommand</code>	91	General: (ASAJ) Split from ltinit.dtx.	49
Replaces <code>\space</code> by ‘ ’ in <code>\csname</code>	91	1994/05/16 ltcntrl.dtx v1.0a	
1994/05/13 ltpictur.dtx v0.1d		General: (ASAJ) Split from ltinit.dtx.	51
General: Removed surplus braces from <code>\@if..</code> constructions	318	1994/05/16 ltdefns.dtx v1.1a	
1994/05/13 lttab.dtx v1.0d		General: (ASAJ) Split from ltinit.dtx.	34
<code>\@contfield</code> : Colour support	302	1994/05/16 lterror.dtx v1.1a	
<code>\@startfield</code> : Colour support	301	General: (ASAJ) Completely new error interface.	55
<code>\@stopfield</code> : Colour support	301	(ASAJ) Split from ltinit.dtx.	55
<code>\a</code> : moved to ltoutenc	300	1994/05/16 ltfinal.dtx v1.0i	
1994/05/14 fontdef.dtx v2.1f		General: moved output enc stuff to lfonts	504
General: Removed .def files.	216		
1994/05/14 ltfsbas.dtx v2.1m			
<code>\enc@update</code> : Macro added	146		
1994/05/14 ltfsbas.dtx v2.1n			
General: Set defaults for all <code>\f@...</code>	147		
<code>\DeclareErrorFont</code> : Don’t set <code>\f@encoding</code>	151		

1994/05/16 ltssbas.dtx v2.1p	Remove \@acci and friends again	29
\fontsize: Pass \baselinestretch not \f@linespread	Remove unnecessary def for \item	28
\linespread: Remove surplus braces	\loop: Use Kabelschacht method	26
1994/05/16 ltssini.dtx v2.1m	\m@th: Remove unnecessary space	28
\acciii: Define saved versions of accents	1994/05/16 ltsspace.dtx v1.1a General: (ASAJ) Split from ltinit.dtx.	66
1994/05/16 ltlogos.dtx v1.1a	1994/05/17 ltclass.dtx v1.0e \@use@ption: Execute option after removing from list, not before	461
General: (ASAJ) Split from ltinit.dtx.	1994/05/17 ltdefs.dtx 1.1b General: (ASAJ) Added the \@protect@... commands.	43
1994/05/16 ltmath.dtx v1.0k	1994/05/17 ltdefs.dtx v1.1b General: (ASAJ) Added definitions for protect.	34
\ensuremath: Use \DeclareRobustCommand and add extra braces in math mode	(ASAJ) Removed warnings and logging to lterror.dtx.	34
1994/05/16 ltoutenc.dtx 1.5h	Added the discussion of protected commands, defined the values that \@protect should have.	43
General: \pounds was still using u rather than ui shape.	1994/05/17 ltdefs.dtx v1.1c General: (ASAJ) Redid definitions for protect.	34
1994/05/16 ltoutenc.dtx v1.5f	1994/05/17 lterror.dtx v1.1b General: (ASAJ) Moved error stuff from ltdefs.dtx.	55
General: enc files now have uc encoding name parts (FMi)	1994/05/17 ltssini.dtx v2.1n \copyright: Really add extra braces	211
Revert code so that the encoding given is used in \DeclareTextCommand (FMi)	\nfss@text: Added braces to allow use in subscripts	211
1994/05/16 ltoutenc.dtx v1.5g	1994/05/17 ltmath.dtx v1.0i General: Replaced \let by \gdef, for indirect definition.	262
General: Made fontenc.sty use the new mixed-case encoding files.	1994/05/17 ltoutenc.dtx v1.5j General: Added braces to \pounds so it works as a subscript.	91
Removed the lowercasing of the filename.	1994/05/18 ltdefs.dtx 1.1c General: (ASAJ) Renamed the commands, and removed one which is no longer needed.	43
1994/05/16 ltoutenc.dtx v1.5h	1994/05/18 ltdefs.dtx v1.1c General: Redid the discussion and definitions, in line with the proposed new setting of \protect in the output routine.	43
General: Added \NG, \ng, \TH, \th, \DH, \dh, \DJ and \dj.	1994/05/18 ltfinal.dtx v0.1j General: Corrected the lccode for d-bar.	496
Added \r (ring accent) and \k (ogonek) accents.		
Fixed a bug with \pounds.		
Removed \P from the OT1 definitions file.		
1994/05/16 ltoutenc.dtx v1.5i		
General: Fixed a bug with \d.		
1994/05/16 ltoutput.dtx v1.0q		
\@writsetup: Changed setting of accents (FMi): with the new encoding setup they can use \let. It could also use the new internal commands?		
General: Changed setting of accents (FMi).		
1994/05/16 ltpar.dtx v1.1a		
General: (ASAJ) Split from ltinit.dtx.		
1994/05/16 ltplain.dtx v1.0h		
General: Comment out encoding specific commands		

1994/05/18 ltlogos.dtx v1.1b	1994/05/20 ltdefs.dtx v1.1e
General: (ASAJ) Added the \TeX	General: Changed command name
logo. 80	from $\backslash@checkcommand$ to
(ASAJ) Made the $\LaTeX 2\epsilon$ logo	$\backslash\checkcommand$ 34
use the text font ‘2’ rather	$\backslash\checkcommand$: Changed name
than the math font ‘2’. 80	from $\backslash@checkcommand$ to
1994/05/18 ltoutenc.dtx v1.5k	$\backslash\checkcommand$ 41
General: Made dotted-i produce ‘i’. 91	1994/05/20 lterror.dtx v1.1c
Removed braces from \backslashpounds	General: (ASAJ) Added
and \backslashdollar 91	$\backslash@latex@info@no@line$ 55
Replaced $\backslashdefaultencoding$	(ASAJ) Added missing full
with $\backslashencodingdefault$ 91	stops. 55
1994/05/19 ltbibl.dtx v1.1a	(ASAJ) Fixed a bug with
General: Initial version of	$\backslash@inmatherr$ 55
ltbibl.dtx, split from	1994/05/20 ltfinal.dtx v0.1l
ltxbib.dtx 376	General: Use new font warning
1994/05/19 ltcounts.dtx v1.1a	commands 501
General: Extracted file from	1994/05/20 ltfloat.dtx v1.0h
ltxcntlen. 132	$\backslash@endfloatbox$: Restore outer
1994/05/19 ltdefs.dtx v1.1d	value of $@nobreak$ switch. 363
General: (RmS) Added definitions	1994/05/20 ltfncmd.dtx v3.3h
for $\backslash@namedef$ and $\backslash@nameuse$	General: Use new error commands 236
again. 34	1994/05/20 ltfssbas.dtx v2.1q
1994/05/19 ltfinal.dtx v0.1k	General: Use new error commands 139
General: Removed \backslashmakeat 496	1994/05/20 ltfsstrc.dtx v2.3i
1994/05/19 ltxglo.dtx v1.1a	General: Use new error command
General: Initial version of	names 159
ltxglo.dtx, split from	1994/05/20 ltmiscen.dtx v1.0l
ltxbib.dtx 374	$\backslash@writefile$: Added correct
1994/05/19 ltxlength.dtx v1.1a	setting of \backslashprotect 251
General: Extract file ltxlength from	1994/05/20 ltmiscen.dtx v1.0m
ltxcntlen. 138	General: Use new warning
1994/05/19 ltxpageno.dtx v1.1a	commands 249
General: Extract file ltxpageno from	1994/05/20 ltoutput.dtx v1.0s
ltxcntlen. 244	$\backslash@writsetup$: Added setting of
1994/05/19 ltxplain.dtx v0.1k ltfinal	\backslashprotect during \backslashshipout 409
\backslashshowoutput : used \backslashmaxdimen not	General: Added setting of
99999 29	\backslashprotect during \backslashshipout 383
\backslashshowoverfull : used $\backslash@one$ not 1 29	1994/05/20 ltxpage.dtx v1.0d
1994/05/19 ltxref.dtx v1.1a	\backslashmarkright : Changed setting for
General: Extract file ltxref from	\backslashprotect 381
ltxcntlen. 245	1994/05/20 ltxsect.dtx v1.0c
1994/05/1g fontdef.dtx v2.1g	General: Correct setting of
General: Removed	\backslashprotect 352
$\backslashDeclareFontEncoding$ for ot1	$\backslashaddcontentsline$: Correct setting
and t1 and input .def files	of \backslashprotect 352
instead 216	1994/05/21 ltbibl.dtx v1.1b
1994/05/2 ltdefs.dtx v1.1f	General: Use new warning
\backslashrenewcommand : Removed surplus	commands 376
\backslashspace in error 39	1994/05/21 lterror.dtx v1.1d
$\backslashrenewenvironment$: Removed	General: (ASAJ) Made the error
surplus \backslashspace in error 40	commands robust. 55
	1994/05/21 ltxfiles.dtx v1.0h
	General: Use new error commands 81

1994/05/21 ltlists.dtx v1.0f		1994/05/23 ltclass.dtx v1.0h	
General: Use new error commands	269	<code>\NeedsTeXFormat</code> : Don't stop	
1994/05/21 ltmiscen.dtx v1.0n		completely when format is	
General: Use new error commands	249	wrong	463
1994/05/21 ltsect.dtx v1.0d		<code>\usepackage</code> : Remove argument if	
General: Use new error commands	344	possible	463
1994/05/21 lttab.dtx v1.0f		1994/05/23 ltdirchk.dtx v1.0f	
General: Use new error commands	296	General: Document <code>\TeXversion</code>	1
1994/05/21 ltxref.dtx v1.1b		1994/05/23 ltfstrc.dtx v2.3j	
General: Use new warning		General: Removed def of	
commands	245	<code>\f@warn@break</code>	176
<code>\newlabel</code> : Use new warning		1994/05/23 ltoutput.dtx v1.0u	
commands	246	<code>\@activechar@info</code> : Added	
1994/05/22 ltclass.dtx v1.0f		<code>\MessageBreak</code>	409
General: Use new warning and		<code>\@writsetup</code> : Changed resetting	
error commands	452	of <code>\protect</code> after shipout to	
1994/05/22 ltdefs.dtx v1.1f		use <code>\aftergroup</code>	409
General: Use new warning and		General: Added <code>\MessageBreak</code>	383
error cmds	34	Changed resetting of <code>\protect</code>	
1994/05/22 lterror.dtx v1.1e		after shipout.	383
General: (ASAJ) Replaced bgroup		1994/05/24 lterror.dtx v1.2e	
by begingroup in error		<code>\@latex@info@no@line</code> : Macro	
messages, to stop extra		added	58
mathords creeping into math		1994/05/24 lterror.dtx v1.2f	
mode.	55	General: (DPC) wrap long lines	55
1994/05/22 lterror.dtx v1.2a		1994/05/24 ltfntcmd.dtx v3.3i	
General: (ASAJ) Made		General: Tidying and typos fixed	236
<code>\GenericError</code> ,		1994/05/24 ltmiscen.dtx v1.0q	
<code>\GenericWarning</code> and		<code>\@currentvline</code> : Use <code>\@empty</code> as	
<code>\GenericInfo</code> robust.	55	outer default	253
(ASAJ) Replaced		1994/05/25 ltdirchk.dtx v1.0g	
<code>\@generic@message</code> and		<code>\filename@parse</code> : Mac parser had	
<code>\@generic@error</code> by		" typo for :	12
<code>\GenericError</code> ,		1994/05/25 ltfntcmd.dtx v3.3j	
<code>\GenericWarning</code> and		General: Insertion of <code>\aftergroups</code>	
<code>\GenericInfo</code> .	55	to implement <code>\nocorr</code> moved	
(ASAJ) Replaced <code>\</code> and tilde		to the end of the group	236
by <code>\MessageBreak</code> and <code>\space</code> .	55	<code>\check@icr</code> : Macros added	239
(ASAJ) Replaces <code>\string</code> by		<code>\check@nocorr@</code> : Insertion of	
<code>\protect</code> in some messages.	55	<code>\aftergroups</code> moved and	
1994/05/22 lterror.dtx v1.2d		defaults set up for efficiency	239
<code>\GenericError</code> : (DPC) Alternative		<code>\DeclareTextFontCommand</code> :	
version added for old TeXs	55	<code>\expandafter</code> inserted	238
(DPC) New version using long		Insertion of <code>\aftergroups</code>	
command name.	55	moved	238
1994/05/22 ltfloat.dtx v1.0i		1994/05/25 ltoutput.dtx v1.0v	
General: Use new warning		General: Extra documentation.	383
commands	354	1994/05/25 ltsect.dtx v1.0e	
1994/05/22 ltoutput.dtx v1.0t		<code>\@dottedtocline</code> : Put braces	
General: Changed warnings and		around argument 4 (the actual	
infos to new commands.	383	toc entry) to avoid font (and	
1994/05/22 ltpictur.dtx v0.1e		possibly other) changes leaking	
General: Use new warning cmds	318	out to the leaders.	353

1994/05/25 ltthm.dtx v1.0c		<code>\check@nocorr@</code> : Added check for empty text	239
General: Modify documentation	340		
1994/05/25 ltvers.dtx v1.0d		1994/06/22 ltfntcmd.dtx v3.3m	
General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here)	32	General: Removed space from <code>\nfss@text</code>	236
1994/05/25 ltxref.dtx v1.1c		Renamed <code>\check@nocorr</code>	236
General: Modify documentation	245	<code>\check@nocorr@</code> : Renamed <code>\check@nocorr</code> to <code>\text@command</code> to improve <code>\long</code> error message	239
1994/05/26 ltfiles.dtx LaTeX2e		<code>\DeclareTextFontCommand</code> : Removed space from <code>\nfss@text</code>	238
<code>\@missingfileerror</code> : Modify message format	88	1994/06/22 ltmath.dtx v1.2t classes	
1994/05/26 ltlogos.dtx v1.1c		<code>\mathindent</code> : Set <code>\mathindent</code> at the end of the class instead of at begin document	267
General: Remove <code>\SLiTeX</code> logo	80	1994/07/20 ltlogos.dtx v1.1e	
1994/05/26 ltplain.dtx v1.1m		<code>\LaTeX</code> : Save a few tokens	80
<code>\iterate</code> : (CAR) added <code>\long</code>	26	<code>\LaTeXe</code> : Save a few tokens	80
<code>\underbar</code> : (CAR/FMi) changed to use box <code>\tw@</code>	28	1994/07/20 ltpage.dtx v1.0h	
1994/05/26 ltplain.dtx v1.1p		<code>\sloppy</code> : Save a few tokens	382
<code>\underbar</code> : (DPC) changed to use <code>\sbox</code>	28	1994/09/16 ltfssbas.dtx v2.1s	
1994/05/26/16 ltmiscen.dtx v1.0r		<code>\nfss@catcodes</code> : Reset [and] as well, just in case	151
General: <code>\literal</code> removed	257	1994/10/07 ltoutenc.dtx v1.5l	
1994/05/29 ltfssdcl.dtx v2.1j		General: Moved the ogonek accent.	91
General: Use new error commands	185	1994/10/11 ltdirchk.dtx v1.0h	
1994/05/31 ltfinal.dtx v1.0n		<code>\@TeXversion</code> : Check for TeX3.14	13
General: Renamed <code>lthyphen.*</code> to <code>lthyphen.*</code>	496	General: Modify all of <code>ltxcheck</code> again	13
1994/06/01 ltboxes.dtx v1.0i		1994/10/12 ltsect.dtx v1.0f	
<code>\@frameb@x</code> : Macro added.	290	General: Doc. typos	344
<code>\@ifframebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code>	289	1994/10/14 fontdef.dtx v2.2a	
<code>\fbox</code> : New version, using <code>\@frameb@x</code>	289	General: New coding	214
<code>\@framebox</code> : New version, so <code>\width</code> is correct in <code>\framebox</code>	289	1994/10/14 ltfssini.dtx v2.2a	
1994/06/01 ltlogos.dtx v1.1d		General: New coding for <code>cfg</code> files	208
<code>\LaTeX</code> : Add <code>\m@th</code> to force math size calculations	80	1994/10/14 ltmiscen.dtx v1.0s	
1994/06/01 ltoutput.dtx v1.0w		General: Move math to other file	249
General: Tidied up typesetting.	383	1994/10/14 ltplain.dtx v1.1a	
1994/06/08 ltfinal.dtx v1.0m		General: Moved code to other files.	14
General: Add patch file system	505	1994/10/15 ltfssbas.dtx v2.1t	
1994/06/09 ltfinal.dtx v1.0n		<code>\extract@alph@from@version</code> : Warn if math alpha is used outside math	156
General: For \TeX 2, do not set codes for higher half of character table.	500, 503	1994/10/18 ltboxes.dtx v1.0j	
1994/06/09 ltfntcmd.dtx v3.3k		<code>\@frameb@x</code> : <code>\leavevmode</code> added	290
General: Tidying and typos fixed in documentation	236	<code>\@ifframebox</code> : <code>\leavevmode</code> moved to <code>\@frameb@x</code>	289
1994/06/18 ltfntcmd.dtx v3.3l		<code>\@parboxto</code> : Macro added to remove misuse of <code>\@empty</code>	291
General: Added check for empty text	236	General: stuff from <code>ltpatch</code> done	285
		<code>\fbox</code> : <code>\long</code> added	289
		<code>\mbox</code> : <code>\long</code> added	286

- `\sbox: \long` added 288
 1994/10/18 `ltclass.dtx v1.0j`
 General: Move `\listfiles` to
 `ltfiles.dtx` 469
 1994/10/18 `ltdefns.dtx v1.2a`
 `\@star@or@long`: macro added . . 37
 General: Add extra test for
 `\endgraf` 34
 Add star-forms for all commands 34
 `\renew@environment`: reset end
 command 40
 1994/10/18 `ltfiles.dtx v1.0i`
 `\listfiles`: code moved here from
 `ltclass` 89
 1994/10/18 `ltoutenc.dtx v1.5l`
 General: Added new definitions of
 `\patterns` and `\hyphenation`. 101
 1994/10/18 `ltoutenc.dtx v1.5m`
 General: Added new definitions of
 `\patterns` and `\hyphenation`. 91
 1994/10/18 `ltsect.dtx v1.0g`
 `\@dottedtocline`: Added
 `\normalcolor` for page
 number 353
 General: Added `\normalcolor` . . 344
 1994/10/19 `lffssbas.dtx v2.1t`
 `\DeclareFontEncoding`: Add
 missing `\relax`. 141
 1994/10/23 `lffsstrc.dtx v23.k`
 `\every@math@size`: Renamed to
 `\every@math@size` 166
 1994/10/23 `ltmath.dtx v1.0l`
 `\@eqnnum`: Added `\normalcolor`
 since `\eqno` introduces a
 subgroup of the displayed math
 group 264
 `\ensuremath`: Remove extra
 braces: but see p 168 of
 Leslie's book 266
 1994/10/24 `ltboxes.dtx v1.0k`
 `\fbox`: Inner braces added (to fix
 `latex/1061`) 289
 1994/10/25 `fontdef.dtx v2.2c`
 General: Added `OMSenc.def` . . . 216
 1994/10/25 `ltboxes.dtx v1.0l`
 `\@isavepicbox`: missing percent
 (moved from `ltpatch`) 288
 1994/10/25 `ltdefns.dtx v1.2b`
 General: Documentation
 improvements 34
 1994/10/25 `ltoutenc.dtx 1.6a`
 General: Added `\textdollar`,
 `\textlbrace`, `\textrbrace`,
 `\textsterling`,
 `\textunderline`. 107
 Removed `\textlbrace`,
 `\textrbrace`, `\textunderline`
 to give them their proper
 names. 107
 1994/10/25 `ltoutenc.dtx v1.6a`
 General: Added
 `\ProvideTextCommand`,
 `\UseTextSymbol`,
 `\UseTextAccent`,
 `\DeclareTextSymbolDefault`,
 `\DeclareTextAccentDefault`,
 `\DeclareTextCommandDefault`,
 and
 `\ProvideTextCommandDefault`. 91
 Added the `\Provide` commands,
 and the default definitions. . . 94
 Added the defaults. 102
 Added the files `OT1enc.def`,
 `T1enc.def` and `OMSenc.def`. . 102
 Added the OMS encoding. . . . 112
 1994/10/27 `ltoutenc.dtx 1.6b`
 General: Added `\textasciicircum`
 `\textasciitilde`
 `\textbackslash` `\textbar`
 `\textbraceleft`
 `\textbraceright`
 `\textcompwordmark`
 `\textemdash` `\textendash`
 `\textexclamdown`
 `\textgreater`
 `\textthyphenchar` `\textthyphen`
 `\textless` `\textquestiondown`
 `\textquotedblleft`
 `\textquotedblright`
 `\textquotedbl`
 `\textquoteleft`
 `\textquoteright`
 `\textunderscore`
 `\textvisiblespace` 107
 Added: `\textemdash`
 `\textendash` `\textexclamdown`
 `\textthyphenchar` `\textthyphen`
 `\textquestiondown`
 `\textquotedblleft`
 `\textquotedblright`
 `\textquoteleft`
 `\textquoteright` 105
 1994/10/27 `ltoutenc.dtx v1.5d`
 General: Rewrote
 `\DeclareTextSymbol` to define
 its argument to use the current
 encoding by default, to fit with

- `\DeclareTextCommand`. 95
- 1994/10/27 ltoutenc.dtx v1.6b
 - General: Added `\textbackslash`. 112
 - Added more defaults for OT1. 102
 - Removed the enc.def files 91
 - Removed the files OT1enc.def, T1enc.def and OMSenc.def. . . 102
 - Renamed `\textlbrace` to `\textbraceleft` and `\textrbrace` to `\textbraceright`. 112
- 1994/10/29 ltmath.dtx 1.0m
 - General: ASAJ: Added `\DeclareMathOperator`. 258
 - ASAJ: Tidied up documentation. 262
- 1994/10/29 ltmath.dtx v1.0m
 - General: ASAJ: Added `\mathellipsis`, `\mathdollar` and `\mathsterling`. 262
 - ASAJ: Removed `\dag`, `\ddag`. 262
 - ASAJ: Renamed `\S` and `\P` to `\mathsection` and `\mathparagraph` and made them `\mathchardefs`. 262
- 1994/10/29 ltoutenc.dtx v1.6c
 - General: Added commands like `\dots` for use in text and math. 102
 - Renamed `\P`, `\S`, `\dag` and `\ddag` to `\textparagraph`, `\textsection`, `\textdagger` and `\textdaggerdbl`. 91
- 1994/10/30 ltdefns.dtx v1.2c
 - `\@onelevel@sanitize`: Macro added 47
 - General: (CAR)`\@onelevel@sanitize` added 34
- 1994/10/30 ltdefns.dtx v1.2f
 - General: (DPC)`\newwrite`'s moved to ltfiles 34
- 1994/10/30 ltmath.dtx v1.0n
 - General: ASAJ: Moved the new commands to ltoutenc. 262
- 1994/10/30 ltoutenc.dtx v1.6d
 - General: Added `\DeclareTextCom-`
`positeCommand`. 91
 - Added `\textcircled`. 91, 103, 112
 - Added `\t`. 103
 - Added math commands. 91
 - Added OML encoding. 91, 103
 - Added the OML encoding. 112
- Made `\textless` and `\textgreater` come from OML. 103
- Moved math commands here from ltmath. 105
- Removed `\textregistered`. 103
- Rewrote `\copyright` to use `\textcircled`. 103
- 1994/10/31 fontdef.dtx v2.2d
 - General: Added OMLenc.def 216
- 1994/10/31 fontdef.dtx v2.2e
 - General: ... and moved further down 216
- 1994/10/31 ltfloat.dtx v1.1a
 - `\@dblfloat`: Major changes since two-column and one-column cases merged 358
 - `\@dblflset`: Macro added 357
 - Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved 357
 - `\@endfloatbox`: (DPC/CAR) Extra box added to remove colour resetting from vmode . 363
 - `\@floatboxreset`: Macro added . 361
 - `\@footnotetext`: (DPC/CAR) Move colour setting to output routine 372
 - `\@savemarbox`: (DPC/CAR) Extra box added for colour 366
 - `\@setfps`: Macro added 358
 - `\@xdblfloat`: Macros removed: `\@dblflt`, `\@xdblfloat` 363
 - `\@xfloat`: (DPC/CAR) Extra box added to remove colour resetting from vmode 359
 - Major changes, removing setting of local variables, space hacks etc; two-column and one-column cases merged 358
 - Reset hook added 359
 - `\@xympar`: (DPC/CAR) Extra box added since needed for floats . 367
 - `\fps@dbl`: Macro added 358
- 1994/10/31 ltoutput.dtx v1.1a
 - `\@makecol`: (DPC/CAR) Colour resetting moved to here 406
 - `\@topnewpage`: (DPC/CAR) Extra box added to remove colour resetting from vmode 398
 - (DPC/CAR) Use `\color@begingroup` for colour . 398

(DPC/CAR) Use	<code>\makeglossary</code> : Removed
<code>\normalcolor</code> 398	<code>\if@filesw</code> from
1994/11/02 ltoutenc.dtx v1.6d	<code>\makeglossary</code> 375
General: Wrapped lines longer	1994/11/04 ltmiscen.dtx v1.0t
than 70 characters. 91	<code>\@writefile</code> : Removed setting of
1994/11/03 ltclass.dtx v1.0k	<code>\protect</code> . ASAJ. 251
General: Move	1994/11/04 ltoutenc.dtx v1.6f
<code>\@missingfileerror</code> to lfiles 456	General: Added <code>_</code> 104
1994/11/03 ltdirchk.dtx v1.0i	Added <code>\mathunderscore</code> 105
General: Generate an error if	1994/11/04 ltpage.dtx v1.0e
<code>latex.ltx</code> not used with clean	<code>\markright</code> : Added
<code>initex</code> 1	<code>\@unexpandable@protect</code> .
1994/11/03 ltfiles.dtx v1.0j	ASAJ. 381
<code>\@missingfileerror</code> : Move here	1994/11/04 ltsect.dtx 1.0h
from ltclass 88	<code>\@sect</code> : (ASAJ) Added
1994/11/04 ltboxes.dtx v1.0m	<code>\protected@edef</code> 347
<code>\@mpfootnotetext</code> : Added	General: (ASAJ) Added
<code>\protected@edef</code> . ASAJ. . . 293	<code>\protected@xdef</code> to <code>\thanks</code> . 344
1994/11/04 ltdefns.dtx v1.2e	1994/11/04 ltsect.dtx v1.0h
General: Added	General: Added <code>\protected@write</code>
<code>\set@display@protect</code> to	to <code>\addtocontents</code> . ASAJ. . . 352
<code>\typeout</code> . ASAJ. 34	<code>\addcontentsline</code> : Added
Added commands for setting	<code>\protected@write</code> to
and restoring <code>\protect</code> . ASAJ. 44	<code>\addcontentsline</code> . ASAJ. . . 352
Rewrote protected short	1994/11/04 lttab.dtx v1.0h
commands using <code>\x@protect</code> .	<code>\@mkpream</code> : (ASAJ) Added
ASAJ. 43	<code>\@unexpandable@protect</code> to
1994/11/04 lterror.dtx v1.2g	<code>\@mkpream</code> 313
General: Added	<code>\multicolumn</code> : (ASAJ) added
<code>\set@display@protect</code> to	<code>\set@typeset@protect</code> 309
<code>\Generic*</code> commands. ASAJ. 55	1994/11/04 ltxref.dtx v1.1d
1994/11/04 ltfiles.dtx v1.0k	<code>\label</code> : (ASAJ)Added
<code>\nofiles</code> : Added setting of	<code>\protected@write</code> 247
<code>\protected@write</code> ,	<code>\refstepcounter</code> : (ASAJ)Added
<code>\makeindex</code> and	<code>\protected@edef</code> 247
<code>\makeglossary</code> to <code>\nofiles</code> .	1994/11/05 ltboxes.dtx v1.0n
ASAJ. 85	<code>\@mpfootnotetext</code> : Colour
<code>\protected@write</code> : Macro added	resetting for footnotes moved
ASAJ. 85	to <code>endminipage</code> : as for main
1994/11/04 ltfloat.dtx v1.1b	page. 293
<code>\@footnotetext</code> : (ASAJ) Added	<code>\color@endbox</code> : macro added for
<code>\protected@edef</code> 372	colour support 287
<code>\footnotemark</code> : Added	<code>\color@hbox</code> : macro added for
<code>\protected@xdef</code> to	colour support 287
<code>\footnotemark</code> 372	<code>\endminipage</code> : Colour resetting for
1994/11/04 ltidxglo.dtx v1.1b	footnotes moved to here: as for
<code>\@wrglossary</code> : Added	main page. 293
<code>\protected@write</code> to	1994/11/05 ltboxes.dtx v1.0o
<code>\@wrglossary</code> 375	<code>\@mpfootnotetext</code> : Colour groups
<code>\@wrindex</code> : Added	restored here. 293
<code>\protected@write</code> to	1994/11/05 ltfloat.dtx v1.1c
<code>\@wrindex</code> 375	<code>\@dblflset</code> : Add compatibility
General: Removed <code>\if@filesw</code>	with old version of <code>\@xfloat</code> . 357
from <code>\makeindex</code> 374	

<code>\@endfloatbox</code> : Use new	(DPC) Updated to use
<code>\color@hbox</code> concept. 363	<code>\ProvidesFile</code> 216
<code>\@footnotetext</code> : Removed	1994/11/07 ltfiles.dtx v1.0l
<code>\normalcolor</code> (again) 372	<code>\@unused</code> : move here from ltdefs, remove duplicate <code>\mainaux</code> . 83
<code>\@savemarbox</code> : Use new	1994/11/07 ltfiles.dtx v1.0m
<code>\color@hbox</code> concept. 366	<code>\document</code> : Renamed <code>\every@size</code> to <code>\every@math@size</code> 84
<code>\@setfps</code> : Add compatibility with old version of <code>\@xfloat</code> 358	1994/11/07 preload.dtx v2.1e
<code>\@xfloat</code> : Add compatibility with old version of <code>\@xfloat</code> : but the arguments, provided at exorbitant cost, are now completely ignored 358	General: (DPC) Updated to use <code>\ProvidesFile</code> 232
Use new <code>\color@hbox</code> concept. 359	1994/11/09 ltboxes.dtx v1.0p
<code>\@xympar</code> : Use new <code>\color@hbox</code> concept. 367	<code>\@finalstrut</code> : Revert <code>\finalstrut</code> to 2.09 equivalent (from ltpatch) 295
1994/11/05 ltoutenc.dtx v1.6g	General: more colour changes. . . . 285
General: Added setting of <code>\@typeset@protect</code> to <code>\patterns</code> and <code>\hyphenation</code> . 101	1994/11/09 ltfssbas.dtx v2.1v
1994/11/05 ltoutput.dtx v1.1b	<code>\@vpt</code> : (DPC) macros added, from setsize.dtx 157
<code>\@topnewpage</code> : Use new	(DPC) reduce save stack usage latex/1742 157
<code>\color@hbox</code> concept. 398	1994/11/10 ltbibl.dtx v1.1c
<code>\@writesetup</code> : Change protect settings for new-style, protect-free aux-files. 409	General: Fix <code>\nocite{*}</code> 376
Use new <code>\color@hbox</code> concept. 409	<code>\nocite</code> : Fix <code>\nocite{*}</code> 378
1994/11/05 ltoutput.dtx v1.1c	1994/11/10 ltmath.dtx v1.2v classes
<code>\@beginDvi</code> : Added macro 412	<code>eqnarray</code> : Added value of <code>\parskip</code> to <code>\abovedisplayskip</code> to compensate for negative <code>\topsep</code> 268
<code>\@beginDvibox</code> : Added macro . . 395	1994/11/10 ltoutput.dtx v1.1e
<code>\@writesetup</code> : Add new	<code>\@writesetup</code> : Modify <code>\protect</code> setting 409
<code>\AtBeginDvi</code> concept 409	1994/11/10 ltplain.dtx v1.1b
<code>\AtBeginDvi</code> : Added macro 395	General: (CAR) added patch to <code>\loop</code> 14
1994/11/06 ltfssbas.dtx v2.1u	<code>\iterate</code> : (CAR) added extra <code>\relax</code> 26
<code>\cf@encoding</code> : New macro 147	1994/11/11 ltspace.dtx v1.2a
<code>\DeclareFixedFont</code> : Renamed	<code>\:</code> (DPC) Make robust 70
<code>\every@size</code> to <code>\every@math@size</code> 140	1994/11/12 ltfntcmd.dtx v3.3o
1994/11/06 ltfssini.dtx v2.2b	<code>\normalsize</code> : Added <code>\MessageBreak</code> 243
<code>\@setszize</code> : Use	1994/11/12 ltlists.dtx v1.2b ltspace
<code>\@typeset@protect</code> 210	<code>\endtrivlist</code> : Changed order of tests to make <code>\@noitemerror</code> correct: end of an era. 278
1994/11/06 ltfsstrc.dtx v2.3k	1994/11/12 ltmiscen.dtx v1.0u
<code>\glb@currszize</code> : New implementation 165	<code>center</code> : Changed end macro to <code>\def</code> : safer and consistent . . 254
<code>\try@simples</code> : New implementation 176	<code>flushleft</code> : Changed end macro to <code>\def</code> : safer and consistent . . 254
<code>\try@size@substitution</code> : New implementation 176	<code>flushright</code> : Changed end macro to <code>\def</code> : safer and consistent 254
<code>\tryis@simple</code> : New implementation 176	
1994/11/07 fontdef.dtx v2.2f	
General: (DPC) Add <code>\DeclareMathSizes</code> declarations 219	

1994/11/12 ltplain.dtx v1.1c	1994/11/17 ltfssstrc.dtx v2.3l
General: Comment out more	General: \@tempa to \reserved@a 159
encoding specific commands 28	1994/11/17 ltmath.dtx v1.0o
1994/11/12 ltspc.dtx v1.2b	General: \@tempa to \reserved@a 258
\addpenalty: Corrected error	1994/11/17 ltmiscen.dtx v1.0v
message 75	General: \@tempa to \reserved@a 249
\advspace: Corrected error	1994/11/17 ltoutenc.dtx v1.6h
message 75	General: (DPC) \@tempa to
1994/11/13 ltspc.dtx v1.2c	\reserved@a 91
\addpenalty: Recorrected error	1994/11/17 ltoutput.dtx v1.1h
message 75	General: \@tempa to \reserved@a. 383
\advspace: Recorrected error	1994/11/17 ltpictur.dtx v1.0f
message 75	General: \@tempa to \reserved@a 318
1994/11/14 ltoutput.dtx v1.1f	1994/11/17 ltsect.dtx v1.0i
\@beginDvi: Use normal box	General: \@tempa to \reserved@a 344
register: why a box? 412	1994/11/17 lttab.dtx v1.0j
\@beginDvibox: Use normal box	General: \@tempa to \reserved@a 296
register: why a box? 395	1994/11/18 ltboxes.dtx v1.0r
\@writesetup: Modify new	\color@vbox: macro added for
\AtBeginDvi concept 409	colour support 287
General: Removed old definition of	1994/11/18 ltfinal.dtx v1.0n
\@testfp. 383	General: re-allow slots 127–255 . . 502
1994/11/14 ltspc.dtx v1.2d	1994/11/18 ltfssbas.dtx v2.1x
\@: (DPC) Macro modified 70	General: (DPC) use \reserved@f
1994/11/14 lttab.dtx v1.0i	not \next 139
\@tabularnewline: (DPC) Macro	1994/11/18 ltfssdcl.dtx v2.1m
added 308	\DeclareMathDelimiter: (DPC)
1994/11/16 fontdef.dtx v2.2h	\expandafter instead of
General: (DPC) Removed \{ and	\next 201
\} 216	1994/11/18 ltfssstrc.dtx v2.3m
1994/11/17 ltboxes.dtx v1.0q	General: \next to \reserved@f . 159
General: \@tempa to \reserved@a 285	1994/11/18 ltmath.dtx v1.0p
1994/11/17 ltclass.dtx v1.0l	\phantom: (DPC) colour support 260
General: \@tempa to \reserved@a 452	(DPC) use \expandafter
1994/11/17 ltcntrl.dtx v1.0b	instead of \next 260
General: \@tempa to \reserved@a 51	\prime@s: (DPC) use \@let@token
1994/11/17 ltdefns.dtx v1.0g	instead of \next and
General: \@tempa to \reserved@a 34	\expandafter instead of \nxt 262
1994/11/17 ltdirchk.dtx v1.0j	\smash: (DPC) colour support . . 260
General: \@tempa to \reserved@a . 1	(DPC) use \expandafter
1994/11/17 lterror.dtx v1.2h	instead of \next 260
General: \@tempa to \reserved@a 55	1994/11/21 ltfloat.dtx v1.1f
1994/11/17 ltfiles.dtx v1.0n	\@endfloatbox: Added reset of
General: \@tempa to \reserved@a 81	minipage flag 363
1994/11/17 ltfinal.dtx v1.0o	Corrected position of
General: \@tempa to \reserved@a 496	\outer@nobreak 363
1994/11/17 ltfloat.dtx v1.1e	\@marginparreset: Macro added 366
General: \@tempa to \reserved@a 354	\@savemarbox: Added
1994/11/17 ltfntcmd.dtx v3.3p	\@setminipage etc 366
General: \@tempa to \reserved@a 236	Added resetting of size and font 366
1994/11/17 ltfssbas.dtx v2.1w	Changed to \color@vbox 366
General: \@tempa to \reserved@a 139	Use \@setnobreak etc 366
1994/11/17 ltfssdcl.dtx v2.1m	\@setminipage: Macro added . . . 361
General: \@tempa to \reserved@a 185	\@setnobreak: Macro added 361

- `\xfloat`: Added `\setminipage` 359
- Added resetting of size and font 359
- Changed to `\color@vbox` so that large floats overflow at the bottom 359
- Missing percents reinserted after 4, 8: these are not numbers. 358
- Use `\setnobreak` 359
- `\xympar`: Changed to `\color@vbox` 367
- 1994/11/21 ltoutput.dtx v1.li
- `\addtocurcol`: Added `\if@nobreak` test before float box 421, 425
- `\specialoutput`: Added `\if@nobreak` test 402
- `\topnewpage`: Changed to `\color@vbox` 398
- 1994/11/22 ltssdcl.dtx v2.1o
- General: wrap long lines 185
- 1994/11/22 ltoutenc.dtx v1.6i
- General: Corrected `\dots` so that there's no kerning in monowidth fonts. 91
- Corrected typo with `\mathunderscore`. 91
- Fixed empty accents. Again. . . 91
- 1994/11/24 ltdefns.dtx v1.2h
- `\newenv`: Added test for `\endgraf` 40
- 1994/11/25 ltplain.dtx v1.1f
- General: (DPC) Comment out lots of obsolete code 14
- 1994/11/26 ltfloat.dtx v1.1b
- `\footnote`: (ASAJ) Added `\protected@xdef`. 371
- 1994/11/28 ltcntrl.dtx v1.0c
- General: Documentation improvements 51
- 1994/11/30 ltfiles.dtx v1.0o
- `\dofilelist`: Macro added 90
- `\listfiles`: Use `\dofilelist` . . 89
- `\nofiles`: There is no `\gobblethree`... 85
- 1994/11/30 ltssbas.dtx v2.1y
- `\fontshape`: Use `\current@cmd` in `\@enc@update`. ASAJ. 146
- 1994/11/30 ltmath.dtx 1.0q
- General: ASAJ: `\DeclareMathOperator` moved to AMS \LaTeX 258
- 1994/11/30 ltmiscen.dtx v1.0w
- `\enddocument`: (DPC) Do warnings even for `\nofiles` . . 251
- (DPC) Use `\dofilelist` 251
- 1994/11/30 ltoutenc.dtx 1.7a
- General: Redefined `\a` for the new scheme. 102
- 1994/11/30 ltoutenc.dtx v1.6g
- General: Removed new definitions of `\patterns` and `\hyphenation`, since encoding-specific commands now expand in the mouth. . . 101
- 1994/11/30 ltoutenc.dtx v1.7a
- General: Added new code for encoding-specific commands. These now expand in the mouth, which means that ligaturing and kerning can happen. 91
- Always load the enc.def file, so that the default encoding for the commands will change. . . 119
- Redefined `\@changed@cmd` to expand in the mouth. 95
- Removed `\@changed@x@mouth` since `\@changed@x` now expands in the mouth. 95
- Rewrote `\@text@composite` so it allows an empty argument, or an argument containing lots of commands. 97
- 1994/12/01 ltfinal.dtx v1.0p
- General: Renamed `lthyphen.*` to `hyphen.*`. 496
- 1994/12/01 lthyphen.dtx v1.0g
- General: Rename `lthyphen.ltx/cfg` to `hyphen.ltx/cfg` 471
- 1994/12/01 ltplain.dtx v1.1g
- General: (DPC) More doc changes 14
- 1994/12/02 fontdef.dtx v2.2i
- General: Commented out `\ldots`. ASAJ. 214
- 1994/12/02 ltssini.dtx v2.2c
- `\copyright`: `\copyright` is now in ltoutenc. ASAJ 211
- 1994/12/02 ltlists.dtx v1.0e
- `\@trivlist`: RmS: Added check for looping 277
- 1994/12/02 ltoutenc.dtx 1.7b
- General: Redefined `\a` properly. . 102
- 1994/12/02 ltoutenc.dtx v1.7b
- General: Fixed a bug with `\a`. . . 91
- 1994/12/04 lthyphen.dtx v1.0h
- General: Documentation edits for /1989 471

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> 91	<code>\copyright</code> so it works
1994/12/06 ltfssbas.dtx v2.1z	unbraced in subscripts. 91
<code>\DeclareFontEncoding:</code> use	Added check for math mode in
<code>\nfss@catcodes</code> 141	<code>\@changed@cmd.</code> 91
<code>\nfss@catcodes:</code> Added tab char	Commented out
as well 151	<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> 91	<code>\textgreater,</code>
1994/12/08 lttab.dtx v1.0k	<code>\textthyphenchar,</code>
<code>\array:</code> Add <code>\tabularnewline</code> . 308	<code>\textthyphen</code> and <code>\textless</code> to
<code>\tabularnewline:</code> (DPC) Made it	save memory. 91
<code>\relax</code> 308	1995/01/12 ltmath.dtx v1.2y classes
1994/12/09 ltbibl.dtx v1.1d	<code>\@eqnnum:</code> Added <code>\normalcolor</code> . 266
<code>\bibliographystyle:</code> (DPC)	1995/03/03 ltoutenc.dtx 1.7g
Allow use in preamble. 378	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 358	tabbing environment. 102
<code>\@dblflset:</code> Macro removed	1995/04/02 ltfntcmd.dtx v3.3r
temporarily 357	<code>\@math@egroup:</code> Read them again
Old version reinstated	to be able to add <code>\relax.</code> . . 243
temporarily 357	1995/04/02 ltfssdcl.dtx v2.1q
<code>\@setfps:</code> Macro removed	<code>\document@select@group:</code> fix
temporarily 358	problem for pr/1275 189
<code>\@xdblfloat:</code> Macros reinserted	<code>\select@group:</code> fix problem for
temporarily 363	pr/1275 187
<code>\@xfloat:</code> Old version reinstated	<code>\set@mathdelimiter:</code> fix pr/1329 204
temporarily 358	1995/04/02 ltfssini.dtx v2.2d
Sanitisation added temporarily 358	<code>\not@math@alphabet:</code> add
General: Some temps reinserted	<code>\noexpand</code> to second part of
temporarily 354	message 209
<code>\fps@dbl:</code> Macro removed	1995/04/21 ltclass.dtx v1.0m
temporarily 358	<code>\DeclareOption*:</code> Made long
1994/12/10 ltfntcmd.dtx v3.3q	/1498 459
<code>\@math@egroup:</code> Don't read	<code>\endfilecontents:</code> Close input
arguments 243	check stream: latex/1487 . . . 467
<code>\check@nocorr@:</code> Use <code>\space</code>	1995/04/21 ltfinal.dtx v1.0q
command for comparison . . . 239	General: Allow initial patch level
1994/12/10 ltfssdcl.dtx v2.1p	0 505
<code>\document@select@group:</code>	1995/04/21 ltoutenc.dtx v1.7h
Surround with braces (add	General: Added <code>\null</code> <code>\k</code>
fourth arg) 189	latex/1274 91
<code>\select@group:</code> Surround with	1995/04/22 ltfiles.dtx v1.0p
braces (add fourth arg) 187	<code>\includeonly:</code> Allow blanks in
1994/12/10 ltoutenc.dtx v1.7e	argument 85
General: Added documentation for	1995/04/22 ltmiscen.dtx v1.0x
the OML encoding. 91	General: Removed extra def of
Replaced width with <code>\@width</code>	<code>\@gobble</code> 249
and ditto height in vrules. . . 91	1995/04/23 ltsect.dtx v1.0j
	<code>\addcontentsline:</code> Use
	<code>\contentsline</code> internally. . . 352

1995/04/24 ltbibl.dtx v1.1e	1995/04/27 ltfiles.dtx v1.0r
\@citex: Add \mbox to undefined case: latex/1239.	\document: Added \global to support groups in hook
377	84
1995/04/24 ltbibl.dtx v1.1f	1995/04/27 ltmiscen.dtx v1.0y
\bibcite: Make \@onlypreamble /1388.	\enddocument: \@checkend moved after hook
377	250
1995/04/24 ltcntrl.dtx v1.0d	1995/04/27 ltplain.dtx v1.1i
\@for: Dont expand second argument with \edef: /1317 (DPC)	General: Move \hang and \textindent to latex209.def
53	28
1995/04/24 ltoutput.dtx v1.1j	1995/04/29 ltcntrl.dtx v1.0e
\fl@tracemessage: Do not add to kernel unless 'trace' specified	General: Moved init of \protect to ltdefs.dtx
438	54
1995/04/24 ltoutput.dtx v1.1l	Removed unused defs for \@setprotect and \@resetprotect
\@beginvibox: Add \vbox latex/1392	54
395	1995/04/29 ltdefs.dtx v1.2j
\@writsetup: Reset \\ latex/1451 (DPC)	\protect: Init \protect here
410	45
1995/04/24 ltpage.dtx v1.0f	1995/04/29 ltpar.dtx v1.1b
\fussy: reset \emergencystretch latex/1344	General: (TO) Comments clean-up.
382	64
1995/04/24 ltplain.dtx v1.1h	1995/05/02 ltsect.dtx v1.0l
\newlanguage: Remove remaining \outer declarations.	\@dottedtocline: Don't reset to \rmfamily
16	353
1995/04/24 ltxref.dtx v1.1e	1995/05/03 ltsect.dtx v1.0m
\newlabel: Make \@onlypreamble for /1388.	General: TO: Promoted documentation to doc.sty standard
246	344
1995/04/25 ltdefs.dtx v1.2i	1995/05/06 ltsect.dtx 1.0n
\@check@c: Make \long for latex/1346	\@secCNTformat: Use \quad instead of \hskip
42	349
\new@environment: Parse arguments slowly but safely /1507	\@sect: Added \relax after \@secCNTformat just in case
40	347
1995/04/25 ltfiles.dtx v1.0q	1995/05/07 ltboxes.dtx v1.0t
\document: Removed execution of \every@size latex/1407	General: Use \hb@xt@
84	285
1995/04/25 ltsect.dtx v1.0k	1995/05/07 ltdefs.dtx v1.2k
\@dottedtocline: Added \hbox around dots.	\hb@xt@: Macro added
353	35
1995/04/27 ltboxes.dtx v1.0s	1995/05/07 ltmath.dtx v1.0r
\@frameb@x: Move \leavevmode for graphics/1512	General: Use \hb@xt@
290	258
\@ifframebox: Move \leavevmode for graphics/1512	1995/05/07 ltoutput.dtx v1.1m
289	General: Use \hb@xt@.
\@iirsbox: Move \leavevmode for graphics/1512	383
295	1995/05/07 ltpictur.dtx v1.0g
\@irsbox: Move \leavevmode for graphics/1512	General: Use \hb@xt@
294	318
\@fbox: Move \leavevmode for graphics/1512	1995/05/07 ltplain.dtx v1.1j
289	General: Use \hb@xt@
\@raisebox: Move \leavevmode for graphics/1512	14
294	1995/05/07 ltsect.dtx v1.0o
	General: Use \hb@xt@
	344
	1995/05/07 lttab.dtx v1.0l
	General: Use \hb@xt@
	296
	1995/05/08 ltbibl.dtx v1.1g
	\@citex: Use \@firstofone
	377
	\bibitem: Removed unnecessary braces
	377
	\@nocite: Use \@firstofone
	378

1995/05/08 ltdefs.dtx v1.2k		1995/05/19 ltpictur.dtx v1.1a	
\typein: Use \@firstofone	36	General: Support autoloading	
1995/05/08 ltdefs.dtx v1.2l		feature	318
\typein: Remove unnecessary		1995/05/20 ltcounts.dtx v1.1b	
braces	36	\@definecounter: Streamlined	
Replace \def by \let	36	code	134
1995/05/08 ltfstrc.dtx v2.3n		\@fnsymbol: Allowing both text	
\ifnot@nil: Use \@firstofone	171	and math	135
1995/05/11 fontdef.dtx v2.2j		\fnsymbol: Streamlined code	134
General: Updates to some plain		1995/05/20 ltcounts.dtx v1.1c	
macros	214	\@definecounter: And do it right	134
1995/05/12 ltclass.dtx v1.0n		1995/05/20 ltfloat.dtx v1.1k	
\DeclareOption*: Use \toks@ to		\@makefnmark: Moved	
remove need to double hash		\normalfont back and use	
/1557	459	\@textsuperscript	370
1995/05/12 ltfloat.dtx v1.1h		Moved \normalfont to	
\@footnotemark: Add \nobreak to		\textsuperscript	370
allow hyphenation. latex/1605	372	\textsuperscript: Use	
1995/05/12 ltpictur.dtx v1.0h		\normalfont.	371
\pictur@: Macro added for		1995/05/21 ltfssdcl.dtx v2.1t	
latex/1355	319	\DeclareMathRadical: Allow for	
1995/05/12 ltvers.dtx v1.0e		undefined cs names	204
General: Add autoload docstrip		1995/05/21 ltlists.dtx v1.0f	
guards	32	General: Moved to doc.sty	
Check for format older than 1		standard	269
year	32	1995/05/21 ltmath.dtx v1.0r	
1995/05/13 ltfstrc.dtx v2.3o		\@sqrt: Use \sqrtsign	264
General: Use single hash mark in		General: Remove \mathhexbox	
\DeclareOption	160	from this file	261
1995/05/16 ltfloat.dtx v1.1i		Update some plain macros	258
\@makefnmark: Now use		\lefteqn: Use \rlap	266
\textsuperscript.	370	\@r@t: Use \sqrtsign instead of	
\textsuperscript: Command		\sqrt	260
added./pr1503	371	1995/05/21 ltoutenc.dtx v1.7h	
\thefootnote: Streamlined parts		\@inmathwarn: Added several	
of code.	370	\@onlypreamble	95
1995/05/17 ltboxes.dtx v1.0u		1995/05/21 ltoutenc.dtx v1.7j	
\@airsbox: Removed surplus		General: Updated some plain	
braces	294	macros	105
1995/05/17 ltclass.dtx v1.0o		1995/05/21 ltplain.dtx v1.1j	
\g@addto@macro: Make long for		General: Moved some code to	
latex/1522	466	other files	14
1995/05/17 ltlists.dtx v1.0g		1995/05/22 ltplain.dtx v1.1k	
\@item: Removed surplus braces	281	General: Definitions of \footins	
\@nbitem: Removed surplus		and \footnoterule moved to	
braces	281	ltfloat.	29
enumerate: Use \thr@@ and		1995/05/22 lttab.dtx v1.1a	
remove surplus braces	282	General: Support autoloading	
itemize: Use \thr@@	283	feature	296
1995/05/18 ltfloat.dtx v1.1j		1995/05/23 ltfssini.dtx v2.2e	
\@makefnmark: Added		\newfont: Font assignment made	
\normalfont.	370	local again.	210
\thempfootnote: Added			
\itshape.	370		

1995/05/24 ltdefs.dtx v1.1l		<code>\InputIfFileExists:</code> (CAR)	
<code>\newif:</code> (DPC) New implementation	41	added <code>\long</code>	88
1995/05/24 ltdefs.dtx v1.2m		<code>\nofiles:</code> (CAR) added <code>\long</code>	85
<code>\typein:</code> (DPC) New implementation	36	<code>\protected@write:</code> (CAR) added <code>\long</code>	85
1995/05/24 ltfloat.dtx v1.1l		1995/05/25 ltfloat.dtx v1.1m	
<code>\@textsuperscript:</code> Command added.	371	<code>\@savemarbox:</code> (CAR) Resettings moved to hook	366
General: Moved definition of <code>\footins</code> and <code>\footnoterule</code> from <code>ltplain</code> .	370	<code>\@xfloat:</code> (CAR) Resettings moved to hook	359
<code>\textsuperscript:</code> Use <code>\@textsuperscript</code>	371	1995/05/25 ltlists.dtx v1.0i	
1995/05/24 ltssbas.dtx v3.0a		<code>\endtrivlist:</code> Macros moved from <code>ltspace.dtx</code>	278
General: (DPC) Make file from previous file, <code>fam.dtx</code>	139	1995/05/25 ltmath.dtx v1.3c classes	
1995/05/20 v2.2d	139	<code>\@eqnnum:</code> replace <code>\reset@font\rmfamily</code> with <code>\normalfont</code> (PR 1578)	266
<code>\mathgroup:</code> (DPC) No need to redefine <code>\newfam</code> as not outer	139	1995/05/25 ltspace.dtx v1.2f	
1995/05/24 ltsscmp.dtx v3.0a		<code>\@vbsphack:</code> (CAR) not used so 'removed'.	74
General: (DPC) Make file from previous file, <code>fam.dtx</code>	181	<code>\@vspacer:</code> (CAR) <code>\@restorepar</code> added to avoid possible infinite tail recursion caused by a typo in the argument.	76
1995/05/24 ltssdcl.dtx v3.0a		(CAR) macros modified to be more efficient	76
General: (DPC) Make file from previous file, <code>latint.dtx</code>	185	General: Macros moved to <code>ltlists.dtx</code>	66
1995/05/24 ltssini.dtx v3.0a		1995/05/26 ltdefs.dtx v1.2n	
General: (DPC) Make file from previous file, <code>lfonts.dtx</code>	208	<code>\@gobblefour:</code> (CAR) Added <code>\longs</code>	42
1995/05/23 v2.2e	213	1995/05/26 ltmath.dtx v1.0s	
<code>\cal:</code> (DPC) Remove definition	213	<code>\@eqnnum:</code> Removed <code>\rmfamily</code> (PR 1578), replaced <code>\reset@font</code> with <code>\normalfont</code>	264
<code>\mit:</code> (DPC) Remove definition	213	1995/05/26 ltpage.dtx v1.0g	
1995/05/24 ltssstrc.dtx v3.0a		<code>\ps@plain:</code> removed <code>\rmfamily</code> (PR 1578)	381
General: (DPC) Make file from previous file, <code>tracefnt</code>	159	1995/05/27 ltssbas.dtx v3.0b	
1995/05/16 v2.3o	159	<code>\mathgroup:</code> (FMi) But a need to define <code>\new@mathgroup</code>	139
1995/05/24 ltssstrc.dtx v3.0b		1995/06/05 fontdef.dtx v2.2k	
General: (DPC) Fix <code>\ProvidesFile</code> usage	159	General: Moved math commands from <code>ltoutenc.dtx</code> .	229
1995/05/25 ltclass.dtx v1.0p		1995/06/05 ltfinal.dtx v1.0r	
<code>\endfilecontents:</code> Delete <code>\filec@ntents</code> after preamble	467	General: Added <code>\MakeUppercase</code> and <code>\MakeLowercase</code> .	496
1995/05/25 ltfiles.dtx v1.0s		1995/06/05 ltoutenc.dtx v1.7k	
<code>\document:</code> Added check for <code>\topskip</code> zero	84	<code>\@inmathwarn:</code> Removed <code>\protected@cmd</code> and replaced with explicit <code>\noexpand</code> .	95
1995/05/25 ltfiles.dtx v1.0t			
<code>\@iffilenamepath:</code> (CAR) added <code>\long</code>	87		
<code>\document:</code> Corrected typo	84		
<code>\IfFileExists:</code> (CAR) added <code>\long</code>	87		

General: Allowed	1995/06/28 ltssini.dtx v3.0b
<code>\ProvideTextCommandDefault</code>	General: (DPC) Fix
after the preamble.	documentation typos
. 97 208
Commented out <code>\textless</code> and	1995/06/28 ltmath.dtx v1.0t
<code>\textgreater</code>	General: minor doc edits
. 103 258
Moved math commands to	1995/07/02 ltplain.dtx v1.1n
<code>fontdef.dtx</code>	General: Removed surplus ‘by’ and
. 105	‘=’ in various places
Save some tokens in 14
<code>\textvisiblespace</code> and	<code>\offinterlineskip</code> : Replaced
<code>\textunderscore</code>	1000 by <code>\@m</code>
. 103 27
1995/06/06 ltfinal.dtx v1.0s	<code>\showoutput</code> : Use <code>\showoverfull</code>
General: Made <code>\MakeUppercase</code>	to save space
and <code>\MakeLowercase</code> brace 29
their argument.	<code>\tracingall</code> : Use <code>\showoutput</code> to
. 496	save space
1995/06/09 ltoutenc.dtx v1.7l 29
<code>\DeclareTextComposite</code> : Rewrote	1995/07/03 ltdefns.dtx v1.2o
<code>\DeclareTextComposite</code> to	<code>\set@typeset@protect</code> : Use
define the composite as a	<code>\@typeset@protect</code> for init
no-argument command rather 45
than a two-argument	1995/07/03 ltfntcmd.dtx v3.3s
command.	<code>\t@st@ic</code> : Use clean interface for
. 98	jump
1995/06/11 ltspac.dtx v1.2g 241
<code>\restorecr</code> : (CAR) <code>\relax</code> added	1995/07/05 ltfntcmd.dtx v3.3s
to stop silent eating of *.	<code>\t@st@ic</code> : Renamed from
. 79	<code>\test@next</code>
1995/06/13 ltfinal.dtx v1.0t 241
General: Add patch level string	1995/07/05 ltspac.dtx v1.2h
more carefully	<code>\@gnewline</code> : Use <code>\break</code>
. 505 70
Call <code>\errorstopmode</code>	<code>\@no@pgbk</code> : Macro replaces <code>\@pgbk</code>
. 506	and <code>\@nopgbk</code>
1995/06/13 ltpictur.dtx v1.1b 69
General: Use <code>\ProvidesFile</code> in	<code>\nopagebreak</code> : Reimplemented
autoload	both using <code>\@no@pgbk</code>
. 318 69
1995/06/14 lttab.dtx v1.1b	1995/07/09 ltcntrl.dtx v1.0f
General: Use <code>\ProvidesFile</code> in	<code>\@iforloop</code> : Reimplemented using
autoload	Kabelschacht method
. 296 53
1995/06/15 ltssbas.dtx v3.0c	<code>\@iwhiledim</code> : Reimplemented
General: (DPC) minor	using Kabelschacht method
documentation changes 52
. 139	<code>\@iwhilenum</code> : Reimplemented
1995/06/15 ltsscmp.dtx v3.0b	using Kabelschacht method
General: (DPC) minor 52
documentation edits	<code>\@iwhilesw</code> : Reimplemented using
. 181	Kabelschacht method
1995/06/15 ltssdcl.dtx v3.0b 52
General: (DPC) minor	<code>\@tfor</code> : Reimplemented using
documentation changes	Kabelschacht method
. 185 54
1995/06/19 ltbibl.dtx v1.1h	1995/07/09 ltlists.dtx v1.0j
<code>\biblecite</code> : Call <code>\@newl@bel</code> so	<code>enumerate</code> : Use <code>\expandafter</code>
repeated keys produce better 282
warning.	<code>itemize</code> : Use <code>\expandafter</code>
. 377 283
1995/06/19 ltclass.dtx v1.0q	1995/07/12 ltpictur.dtx v1.1d
<code>\documentclass</code> : Dont redefine	General: allow 2e commands in 209
<code>\usepackage</code> in compat mode	mode. latex/1737
for /1634 318
. 462	1995/07/13 ltdefns.dtx v1.0p
1995/06/19 ltxref.dtx v1.1e	General: Updates to
<code>\newlabel</code> : Use <code>\@newl@bel</code> to	documentation
share code with <code>\biblecite</code> 34
. 246	1995/07/13 ltfiles.dtx v1.0u
	General: Updates to docu
 81
	1995/07/13 ltssbas.dtx v3.0d
	<code>\@@defaultsubs</code> : macro added
 154
	<code>\@defaultsubs</code> : macro added
 154

General: minor documentation changes	139	1995/08/16 ltfiles.dtx v1.0v	
<code>\wrong@fontshape</code> : Change a macro not a switch to flag default font substitutions . . .	153	<code>\document</code> : set <code>\@maxdepth</code>	84
1995/07/13 ltmiscen.dtx v1.0z		set <code>\do</code> globally	84
<code>\@centercr</code> : Use <code>\nobreak</code>	254	set <code>\topskip</code> globally	84
<code>\@writefile</code> : Added missing percent and use <code>\relax</code> in the THEN case	251	1995/08/24 ltfssbas.dtx v3.0f	
<code>\@xobeysp</code> : Use <code>\nobreak</code>	255	General: Added autoload code . .	139
General: Improve Documentation	249	1995/08/24 ltfstrc.dtx v3.0c	
<code>\enddocument</code> : Set <code>\@setckpt</code> to <code>\@gobbletwo</code> instead of defining it by hand	250	General: Macro	
Shorten redefinition of <code>\bibcite</code> and <code>\newlabel</code>	250	<code>\gobble@font@spec</code> removed	171
Use <code>\@defaultsubs</code> instead of switch	251	<code>\tryis@simple</code> :	178
1995/07/14 ltbibl.dtx v1.1i		1995/08/25 ltoutput.dtx v1.1p	
<code>\bibcite</code> : Remove <code>\@onlypreamble</code> so still defined in new <code>\enddocument</code>	377	General: Support autoloading feature (FMI).	383
1995/07/14 ltxref.dtx v1.1g		1995/09/01 lterror.dtx v1.2i	
<code>\newlabel</code> : Remove <code>\@onlypreamble</code> so still defined in new <code>\enddocument</code>	246	General: Add autoload support . .	55
1995/07/19 ltfssini.dtx v3.0d		1995/09/01 ltplain.dtx v1.1m	
General: (DPC) TeX2 support . .	212	<code>\empty</code> : Use <code>\let</code> to save space . .	26
1995/07/20 ltboxes.dtx v1.0v		<code>\I</code> : Use <code>\let</code> to save space	26
<code>\@isavebox</code> : Use <code>\sbox</code>	288	1995/09/14 ltplain.dtx v1.1o	
<code>\@isavepicbox</code> : Use <code>\sbox</code>	288	General: Moved <code>\multispan</code> to <code>lftab.dtx</code>	14
1995/07/21 ltoutput.dtx v1.1o		1995/09/14 lftab.dtx v1.1c	
<code>\@writesetup</code> : Command added .	409	<code>\cline</code> : (DPC) New implementation	316
New, experimental, versions: need in-lining	409	1995/09/15 ltfssini.dtx v3.0e	
1995/08/09 ltmath.dtx v1.0u		General: (DPC) Modify TeX2 message	212
General: Added code for class options <code>leqno</code> and <code>fleqn</code>	266	1995/09/19 ltmiscen.dtx v1.1a	
1995/08/11 ltlength.dtx v1.1b		<code>\verb</code> : Put <code>\@noligs</code> after <code>\verbatim@font</code> where it belongs.	256
General: Doc typos fixed for latex/753	138	1995/10/01 ltfiles.dtx LaTeX2e	
1995/08/16 ltcntrl.dtx v1.0g		<code>\@addtofilelist</code> : Macro added . .	89
<code>\@break@tfor</code> : Made long	54	1995/10/02 ltdefns.dtx v1.2q	
<code>\@forloop</code> : Made defs long	53	<code>\@ifnch</code> : Use <code>\@let@token</code> for internal/924, save <code>\reserved@e</code>	47
<code>\@fornoop</code> : Made defs long	53	<code>\@ifnextchar</code> : Use <code>\@let@token</code>	46
<code>\@iforloop</code> : Made defs long	53	<code>\@protected@testopt</code> : Macro added	38
<code>\@iwhiledim</code> : Made defs long . . .	52	<code>\@testopt</code> : Macro added	38
Removed <code>\@whilenoop</code>	52	<code>\@xargdef</code> : New implementation, using <code>\@test@opt</code>	37
<code>\@iwhilenum</code> : Made defs long . . .	52	1995/10/02 ltplain.dtx v1.1p	
Removed <code>\@whilenoop</code>	52	General: Move <code>\newif</code> to <code>ltdefns</code> . .	22
<code>\@iwhilesw</code> : Removed		1995/10/03 fontdef.dtx v2.2l	
<code>\@whileswnoop</code>	52	General: <code>\@@sqrt</code> from patch file for /1701	214
<code>\@tfor</code> : Made defs long	54	1995/10/03 ltdefns.dtx v1.2r	
		<code>\@typein</code> : Add missing <code>\@typein</code> for /1710 (from patch file) . . .	36
		1995/10/03 ltpictur.dtx v1.1e	
		General: New autoload code	318
		1995/10/04 ltfssbas.dtx v3.0g	
		General: Modify autoload code . .	139

1995/10/04 ltfssstrc.dtx v3.0d	<code>\nopagebreak:</code> (DPC) Use	
General: (DPC) Modify autoloading code	<code>\@testopt /1911</code>	69
1995/10/04 lttab.dtx v1.1d	1995/10/16 ltthm.dtx v1.0g	
General: Modify autoloading support	General: Revert to previous <code>\newtheorem</code> behaviour	340
1995/10/06 ltfiles.dtx v1.0w	1995/10/17 ltclass.dtx v1.0r	
<code>\@missingfileerror:</code> Autoloading error	<code>\@providesfile:</code> Delay definition of <code>\ProvidesFile</code> till <code>ltfinal</code>	459
1995/10/09 lterror.dtx v1.2j	<code>\ProcessOptions*</code> : Reset <code>\CurrentOption</code> for graphics/1873	461
General: Modify autoloading support	1995/10/17 ltdirchk.dtx v1.0l	
1995/10/09 ltoutenc.dtx v1.7m	General: Modify initex version of <code>\ProvidesFile</code>	4
<code>\@inmathwarn:</code> Autoloading error	1995/10/17 ltfinal.dtx v1.0v	
1995/10/10 ltfssbas.dtx v3.0h	<code>\@providesfile:</code> reset macro	506
<code>\showhyphens:</code> Use <code>\normalfont</code> and make colour safe, and autoloading	<code>\reserved@b:</code> reset here after the <code>\input</code> above	505
1995/10/10 ltfssdcl.dtx v3.0c	1995/10/17 ltplain.dtx v1.1s	
<code>\non@alpherr:</code> (DPC) autoloading error message	<code>\eject:</code> Move <code>\supereject</code> to compat file	27
1995/10/10 ltplain.dtx v1.1r	1995/10/17 lttab.dtx v1.1e	
General: Autoloading tracing code	<code>\@cline:</code> (DPC) Use <code>\@multicnt</code>	316
1995/10/10 ltthm.dtx v1.0f	<code>\@multispan:</code> (DPC) Macro added.	316
General: Make <code>\newtheorem</code> ‘only preamble’	1995/10/19 ltfinal.dtx v1.0w	
1995/10/11 ltoutput.dtx v1.1r	<code>\@filelist:</code> Move after <code>\reserved@a</code> setting:-)	506
<code>\clearpage:</code> Added a check so that it does not lose the argument of <code>\twocolumn[...]</code>	1995/10/20 ltbibl.dtx v1.1k	
1995/10/16 ltbibl.dtx v1.1j	<code>\@citex:</code> Removed redefined flag	377
<code>\cite:</code> (DPC) Make robust	<code>\nocite:</code> Removed redefined flag	378
1995/10/16 ltboxes.dtx v1.0w	1995/10/20 ltclass.dtx v1.0s	
General: Clarify makebox description	<code>\@begindocumenthook:</code> Make setting conditional, for autoloading version	466
1995/10/16 ltdefs.dtx v1.2u	1995/10/20 ltfssbas.dtx v3.0i	
<code>\@ifstar:</code> (DPC) New implementation, for /1910	General: (DPC) Modify autoloading code, change <code>\undefined</code>	139
<code>\new@command:</code> (DPC) Use <code>\@testopt /1911</code>	1995/10/20 ltfssstrc.dtx v3.0e	
<code>\@testopt /1911</code>	General: (DPC) Modify autoloading code	159
<code>\new@environment:</code> (DPC) Use <code>\@testopt /1911</code>	1995/10/22 ltfssbas.dtx v3.0j	
<code>\typein:</code> (DPC) Use <code>\@testopt /1911</code>	General: (RmS) New size function macro <code>\genb@sfcnt</code> needs to be disabled at <code>\document.</code>	139
1995/10/16 ltfssini.dtx v3.0f	1995/10/22 ltfssstrc.dtx v3.0f	
<code>\p@reset@font:</code> Added <code>\relax</code> after <code>\usefont</code> , as the latter eats up spaces.	General: Added ‘genb’ and ‘sgenb’ size functions to support new DC font naming scheme.	159
1995/10/16 ltmath.dtx v1.0y	1995/10/23 lttab.dtx v1.1f	
<code>\@yeqncr:</code> (DPC) Use <code>\@testopt /1911</code>	<code>\@settab:</code> (CAR) Ensure that <code>\@hightab</code> increases by at most	
<code>\sqrt:</code> (DPC) Make robust /1808		
1995/10/16 ltspc.dtx v1.2j		
<code>\nolinebreak:</code> (DPC) Use <code>\@testopt /1911</code>		

one	303	1995/10/27 ltpictur.dtx v1.1f	
<code>\@startline</code> : (CAR)Ensure that		General: Move initialisation to	
<code>\@nxttabmar</code> is never larger		kernel from autoloading file	337
than <code>\@hightab</code>	301	1995/10/31 ltboxes.dtx v1.0x	
<code>\poptabs</code> : (CAR)Ensure that		<code>\@finalstrut</code> : Add <code>\nobreak</code> in	
<code>\@curtab</code> is never larger than		horiz mode to allow	
<code>\@hightab</code>	304	hyphenation. internal/1931	295
<code>\tabbing</code> : (CAR)Make <code>\@hightab</code>		1995/11/01 fontdef.dtx v2.2m	
consistently a local variable	302	General: add <code>\nfss@catcodes</code> for	
1995/10/24 ltfiles.dtx v1.1a		internal/1932	217
<code>\document</code> : Removed		1995/11/01 ltdirchk.dtx v1.0n	
<code>multiplelabels</code> switch	83	General: Initialise	
Removed <code>refundefined</code> switch	84	<code>\@addtofilelist</code> to <code>\@gobble</code>	4
1995/10/24 ltfssbas.dtx v3.0k		1995/11/01 ltfinal.dtx v1.0x	
<code>\@defaultsubs</code> : macro removed	154	General: (DPC) Switch meaning of	
<code>\wrong@fontshape</code> : Make this code		<code>\@addtofilelist</code> for cfg files	501
inline since it happens only		1995/11/01 ltfssbas.dtx v3.0m	
here	153	<code>\DeclareFontShape</code> : (DPC) Test	
1995/10/24 ltmiscen.dtx v1.1b		for <code>\relax</code> not <code>\undefined</code> ,	
<code>\enddocument</code> : Changed logic for		internal/1933	140
producing warning messages		1995/11/01 ltfssini.dtx v3.0g	
and removed switch	251	General: (DPC) Switch meaning of	
Use <code>\@refundefined</code> instead of		<code>\@addtofilelist</code> for cfg files	212
switch	251	1995/11/02 ltfssbas.dtx v3.0n	
1995/10/24 ltxref.dtx v1.1h		<code>\wrong@fontshape</code> : (DPC)	
<code>\@multiplelabels</code> : Switch for		Remove extra space with	
<code>multiplelabels</code> removed	247	<code>\string</code> for latex/1676	152
<code>\@newl@bel</code> : Switch for		1995/11/02 ltoutenc.dtx v1.7n	
<code>multiplelabels</code> replaced by		General: Changed internal name	
inline code	246	<code>\a</code> to <code>\@tabacckludge</code> to	
<code>\@refundefined</code> : Switch for		protect against redefinition by	
<code>refundefined</code> replaced	246	malicious users.	102
<code>\@setref</code> : Switch for <code>refundefined</code>		1995/11/07 ltlists.dtx v1.0k	
renamed	246	<code>\@doendpe</code> : Enclosed <code>\setbox0</code>	
<code>\if@multiplelabels</code> : Macro		assignment by a group so that	
removed	247	it leaves the contents of box 0	
1995/10/25 ltalloc.dtx v1.1b		intact.	279
General: General doc		1995/11/07 ltoutenc.dtx v1.7o	
improvements	49	General: Added <code>\leavevmode</code> at	
1995/10/25 ltfloat.dtx v1.1n		start of <code>\c</code> , otherwise the	
<code>\@endfloatbox</code> : (CAR) macro		output routine might be	
added: to unify code for double		invoked within the macro.	105
and single versions	363	Changed <code>\char32</code> to <code>\@xxxii</code>	
<code>\end@dblfloat</code> : (CAR) unify code		(two tokens less).	106
for double and single versions	362	Replaced octal number 27 by	
<code>\end@float</code> : (CAR) unify code for		decimal number 23 to protect	
double and single versions	361	against the quote character	
1995/10/25 ltidxglo.dtx v1.1d		being active.	106
General: Doc cleanup	374	Replaced some 0's by <code>\z@</code>	
1995/10/25 ltsect.dtx v1.0q		(faster).	106
<code>\subparagraphmark</code> : Use <code>\let</code> not		1995/11/10 ltoutput.dtx v1.1s	
<code>\def</code> to save space.	351	<code>\@shipoutsetup</code> : Command	
		removed	409

<code>\@writsetup</code> : Command removed	409	Added <code>\textless</code> and <code>\textgreater</code>	103, 113
1995/11/14 <code>ltclass.dtx</code> v1.0t <code>\@unprocessedoptions</code> : Allow empty option	409	1995/12/01 <code>ltoutenc.dtx</code> v1.7u General: Made <code>\SS</code> a Default, rather than having the default point to the OT1 definition.	103
<code>\loadwithoptions</code> : macro added	462	1995/12/04 <code>ltspace.dtx</code> v1.2k <code>\nobreakspace</code> : (Macro added	78
<code>\LoadClassWithOptions</code> : macro added	462	1995/12/04 <code>ltspace.dtx</code> v1.2l <code>\xobeysp</code> : (braces added to definition of tilde	78
<code>\RequirePackageWithOptions</code> : macro added	462	1995/12/04 <code>preload.dtx</code> v2.4e General: Ulrik Vieth. added 12pt OMS and OML preloads /1989	234
1995/11/17 <code>ltfssbas.dtx</code> v3.0m <code>\@wrong@font@char</code> : (DPC) Macro added. <code>latex/1676</code>	154	1995/12/05 <code>ltdefns.dtx</code> 1.2w <code>\@unexpandable@noexpand</code> : Removed as never used. internal/1733	43
<code>\define@newfont</code> : Redefine <code>\typeout latex/1676</code>	149	1995/12/05 <code>ltfiles.dtx</code> v1.1c <code>\document</code> : <code>\ignorespaces</code> added for <code>latex/1933</code>	85
<code>\wrong@fontshape</code> : Support <code>\@wrong@font@char latex/1676</code>	152	1995/12/05 <code>lfloat.dtx</code> v1.1n <code>\@textsuperscript</code> : Use <code>\ensuremath</code> for <code>latex/1984</code>	371
1995/11/17 <code>ltoutenc.dtx</code> v1.7p <code>\UseTextSymbol</code> : Support <code>\@wrong@font@char latex/1676</code>	99	1995/12/05 <code>ltoutenc.dtx</code> v1.7v <code>\@inmathwarn</code> : Changed <code>\TextSymbolUnavailable</code> text	96
1995/11/18 <code>ltoutenc.dtx</code> v1.7q <code>\UseTextSymbol</code> : Modify message slightly	99	1995/12/06 <code>ltfssbas.dtx</code> v3.00 <code>\nfss@catcodes</code> : Reset hat, for typeouts etc in fd files	151
1995/11/21 <code>fontdef.dtx</code> v2.2n General: Incorporate changed figures, as in <code>plain.tex</code>	228	1995/12/07 <code>ltbibl.dtx</code> v1.1l <code>\@citex</code> : Restored name of <code>\G@refundefinedtrue</code>	377
1995/11/27 <code>ltfssbas.dtx</code> v3.0n <code>\nfss@catcodes</code> : Reset hash, for definitions in fd files	151	1995/12/07 <code>lfloat.dtx</code> v1.1m <code>\@textsuperscript</code> : Move <code>\m@th</code> out of the <code>\ensuremath</code> for <code>latex/1984</code>	371
1995/11/28 <code>lfloat.dtx</code> v1.1n General: documentation fixes	354	1995/12/07 <code>ltxref.dtx</code> v1.1i <code>\@setref</code> : Switch for redefined restored	246
1995/11/28 <code>lfsstrc.dtx</code> v3.0g General: documentation fixes	159	<code>\G@refundefinedtrue</code> : Renamed (back) from <code>\G@refundefined</code>	246
1995/11/28 <code>ltoutenc.dtx</code> v1.7r General: Added math mode checks to text commands.	95	1995/12/11 <code>ltoutenc.dtx</code> v1.7w General: Modified <code>\copyright</code>	103
doc fixes	91	1995/12/13 <code>ltdefns.dtx</code> 1.2x <code>\-</code> : Documentation changed.	34
Renamed <code>\@changed@x@err</code> to <code>\TextSymbolUnavailable</code>	95	1996/01/10 <code>ltfiles.dtx</code> v1.1d <code>\@iffileonpath</code> : Change argument handling to not require doubled hash. <code>latex/2024</code>	87
1995/11/29 <code>ltoutenc.dtx</code> v1.7t General: Added <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textbackslash</code> , <code>\textbar</code> , <code>\textgreater</code> and <code>\textless</code>	108		
Added <code>\textasciicircum</code> , <code>\textasciitilde</code> , <code>\textregistered</code> and <code>\texttrademark</code>	103		
Added <code>\textbackslash</code> and <code>\textbar</code>	103, 112		

1996/01/20 ltidxglo.dtx v1.1e	(DPC) Moved brace to allow commands like
<code>\makeglossary</code> : Make no-op after use pr/2048 375	<code>\MakeUppercase</code> in 6th argument. Changed <code>\par</code> to <code>\endgraf</code> to allow non-long commands. internal/2148 . . 347
<code>\makeindex</code> : Make no-op after use pr/2048 375	<code>\@ssect</code> : (DPC) Added extra braces for internal/2148 . . . 350
1996/01/20 ltspace.dtx v1.2m	(DPC) Moved brace to allow commands like
<code>\vspace</code> : Made robust 76	<code>\MakeUppercase</code> in 4th argument. Changed <code>\par</code> to <code>\endgraf</code> to allow non-long commands. internal/2148 . . 350
1996/03/25 ltmath.dtx v1.1a	1996/05/23 ltoutenc.dtx v1.7z
<code>\@ensuredmath</code> : Macro added for amslatex/2104 266	<code>\@strip@args</code> : <code>\expandafter</code> added to match other changes for latex/2133 99
<code>\ensuremath</code> : Reimplement for amslatex/2104 266	<code>\add@accent</code> : macro added. latex/2133 97
1996/04/18 ltpage.dtx v1.0i	<code>\DeclareTextAccent</code> : Reimplemented using <code>\add@accent</code> to save space latex/2133 97
General: Improve documentation 380	<code>\DeclareTextCompositeCommand</code> : Modified to cope with new <code>\add@accent</code> command: required removal of check for one argument-command 98
1996/04/22 ltmiscen.dtx v1.1c	1996/05/24 ltoutput.dtx v1.1t
General: Improve Documentation 249	<code>\@specialoutput</code> : Check that <code>\@colroom</code> is less than <code>\vsize</code> , indicating that a float has been added 400
1996/04/22 ltspace.dtx v1.2n	Cut-off point changed to <code>1.5\baselineskip</code> 400
General: Documentation	<code>\@topnewpage</code> : Cut-off point changed to <code>2.5\baselineskip</code> 399
Improvements 66	1996/05/25 ltoutput.dtx v1.1u
1996/04/22 lttab.dtx v1.1g	<code>\@specialoutput</code> : Correct the above check 400
<code>\@tabclassz</code> : (DPC) Extra <code>\hskip</code> keeps <code>tabcolsep</code> in empty columns internal/2122 314	1996/06/03 ltmiscen.dtx v1.1d
1996/04/23 ltcounsts.dtx v1.1d	<code>\@verbatim</code> : Exchanged the following two code lines so that <code>\dospecials</code> cannot reset the category code of characters handled by <code>\@noligs</code> 255
General: Documentation improvements 132	General: Move setting of <code>verbatim</code> font and <code>\@noligs</code> 249
1996/04/24 ltfiles.dtx v1.1e	<code>\verb</code> : Put setting of <code>verbatim</code> font after <code>\dospecials</code> so that <code>\dospecials</code> cannot reset the category code of characters handled by <code>\@noligs</code> 256
<code>\AtBeginDocument</code> : Reset <code>\AtBeginDocument</code> eg for latex/1297 84	
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!! 169	
1996/05/09 ltfntcmd.dtx v3.3t	
<code>\check@icr</code> : Default definitions added 239	
1996/05/17 fontdef.dtx v2.2o	
General: <code>\@@sqrt</code> removed, at last 214, 227	
1996/05/17 ltfiles.dtx v1.1f	
<code>\nofiles</code> : added <code>\write</code> to <code>\protected@write</code> for latex/2146 85	
1996/05/18 ltoutenc.dtx v1.7x	
General: Produce error if encoding not found. pr/2054 119	
1996/05/21 ltoutenc.dtx v1.7y	
General: Corrected error message (CAR) 119	
1996/05/21 ltsect.dtx v1.0s	
<code>\@sect</code> : (DPC) Added extra braces for internal/2148 348	

1996/06/10 ltboxes.dtx v1.0y	1996/07/26 ltfsbas.dtx v3.0p
\@parboxto: (DPC) Changed	\@DeclareMathSizes: use faster
\endgraf to \@@par 291	\if test 144
1996/06/10 ltsect.dtx v1.0t	\nfss@catcodes: omit \relax as
\@sect: (DPC) Changed \endgraf	not needed 151
to \@@par 347	1996/07/26 ltssdcl.dtx v3.0e
\@ssect: (DPC) Changed	\init@restore@version: Removed
\endgraf to \@@par 350	\ifrestore@version switch
1996/06/13 ltdirchk.dtx v1.0r	and replaced by
General: documentation	\init@restore@version . . . 189
improvements mainly from	1996/07/26 ltfsstrc.dtx v3.0i
internal/2174 1	\init@restore@glb@settings:
1996/06/14 lttab.dtx v1.1h	macro added replacing
\@tabclassz: (DPC) Change	\if@inmath switch 168
both\z@skip to lsp for	1996/07/26 ltlists.dtx v1.0l
latex/2160 314	\@item: Remove unecessary
1996/06/22 ltspace.dtx v1.2o	\global before
General: Documentation of	\@minipage... 280
problems added 66	Remove unecessary \global
1996/07/10 ltfinal.dtx v1.0y	before \@nobreak... 281
\@toks: Free up memory from	1996/07/26 ltmath.dtx v1.1b
scratch registers /2213 506	General: Removed \global before
1996/07/19 ltoutenc.dtx v1.8a	\@ignoretrue in various
\@strip@args: Use char 0 not @ as	places. 258
carrier for \lowercase /2197 . 99	1996/07/26 ltmiscen.dtx v1.1e
1996/07/26 ltboxes.dtx v1.0z	\@ignorefalse: put \global into
\if@minipage: put \global into	definition 250
definition 292	\begin: remove \global before
1996/07/26 ltclass.dtx v1.0u	\@ignore... 253
\@classoptionslist: made only	\end: remove \global before
preamble 456	\@ignore... 253
\@unusedoptionlist: made only	\ignorespacesafterend: user level
preamble 456	macro added 250
1996/07/26 ltdefs.dtx v1.2y	1996/07/26 ltoutput.dtx v1.1v
\@reargdef: third arg picked up	\@testfp: remove \global before
by \@yargdef 39	\@test... 441
\@renew@command: use \noexpand	\@xtryfc: remove \global before
instead of \string 39	\@test... 415
use \relax in place of empty arg 39	\@ztryfc: remove \global before
\@renew@environment: use \relax	\@test... 417
in place of empty arg 40	\clearpage: add number of
1996/07/26 ltfloat.dtx v1.1n	missing percents 397
\@endfloatbox: remove unecessary	1996/07/26 ltplain.dtx v1.1t
\global before	\sh@ft: replace \dimen\z@ by
\@minipage... 363	\dimen@ 28
\@savemarbox: remove unecessary	1996/07/26 ltsect.dtx v1.0u
\global before	\@starttoc: removed \global
\@minipage... 366	before \@nobreak... 352
\@setminipage: remove unecessary	\@xsect: Removed \global before
\global before	\@nobreak... 349
\@minipage... 361	1996/07/26 ltspace.dtx v1.2p
\@setnobreak: remove unecessary	\if@nobreak: put \global inside
\global before \@nobreak... 361	definition 71

1996/07/27 ltssbas.dtx v3.0q		1996/10/05 ltfiles.dtx v1.1h	
General: <code>\if@inmath</code> switch		<code>\clubpenalty</code> : Added setting its	
removed	148	value	83
1996/07/27 ltspac.dtx v1.2q		1996/10/08 ltfntcmd.dtx v3.3u	
General: Further documentation of		<code>\DeclareTextFontCommand</code> :	
problems	66	Removed <code>\check@icr</code> when in	
1996/07/27 ltspac.dtx v1.2r		vmode since it causes various	
General: Correct documentation of		errors (see pr/2157)	238
problems	66	1996/10/21 lttab.dtx v1.1i	
1996/08/02 ltfloat.dtx v1.1o		<code>\@array</code> : Use	
<code>\xympar</code> : Remove <code>\global</code> before		<code>\set@typeset@protect</code>	308
<code>\@ignore</code>	367	General: Moved the code	
1996/08/02 ltsect.dtx v1.0v		associated with <code>\mkpream</code> into	
<code>\@afterheading</code> : Removed		the group provided by the box,	
<code>\global</code> before <code>\@nobreak</code>	350	for robustness (latex/2183)	307
1996/08/02 ltspac.dtx v1.2s		<code>\multicolumn</code> : Make	
<code>\@Eshpack</code> : Remove <code>\global</code>		<code>\multicolumn</code> long	
before <code>\@ignore</code>	73	(latex/2180)	309
1996/08/25 ltssbas.dtx v3.0r		<code>\tabbing</code> : Moved the <code>\indent</code> so	
<code>\nfss@catcodes</code> : Reset the acute,		that the <code>\everypar</code> can remove	
grave and double quote chars		it when necessary; this is	
as well	151	needed because the code for	
1996/09/21 ltoutput.dtx v1.1w		items in lists has changed (see	
<code>\@writesetup</code> : Added		pr/22111)	302
<code>\@parboxrestore</code> and made		1996/10/23 ltlists.dtx v1.0m	
consequent deletions: wait for		<code>\@item</code> : <code>\@nobreak</code> moved into	
the howls of protest	409	the <code>\everypar</code> and not	
1996/09/25 ltdirchk.dtx v1.0t		executed unconditionally, see	
General: Move <code>ltxcheck</code> to separate		above	281
file	13	<code>\kern</code> changed to	
1996/09/28 ltmiscen.dtx v1.1f		<code>\setbox</code>	280
<code>\@xobeysp</code> : Moved to <code>ltspac.dtx</code>	255	Added setting of <code>\clubpenalty</code>	
1996/09/28 ltspac.dtx v1.2t		and set <code>\@nobreakfalse</code> only	
<code>\@xobeysp</code> : Moved from		when necessary	281
<code>ltmiscen.dtx</code> and redefined to		1996/10/23 ltsect.dtx v1.0x	
use <code>\nobreakspace</code>	78	<code>\@xsect</code> : Replaced <code>\hskip</code>	
1996/09/29 ltfiles.dtx v1.1g		with <code>\setbox</code> as used in	
<code>\document</code> : Added disabling of		<code>\@afterheading</code>	349
<code>\@nodocument</code>	85	1996/10/24 ltboxes.dtx v1.1a	
1996/09/29 ltoutput.dtx v1.1x		<code>\@arrayparboxrestore</code> : Added	
<code>\newpage</code> : Checks for <code>noskipsec</code>		local settings of flags:	
and <code>inlabel</code> added	397	dangerous!	291
1996/09/29 ltsect.dtx 1.0w		<code>\@iiiminipage</code> : Use it or lose it	
<code>\@noskipsectrue</code> : Added		(<code>@setminpage</code>): Frank will	
documentation	345	want to lose it	292
1996/09/30 ltoutput.dtx v1.1y		1996/10/24 ltfloat.dtx v1.1p	
<code>\newpage</code> : Checks for <code>noskipsec</code>		<code>\@floatboxreset</code> : Added local	
and <code>inlabel</code> removed pending		settings of flags: dangerous!	361
further tests	397	<code>\@marginparreset</code> : Added local	
1996/10/04 ltclass.dtx v1.0v		settings of flags: dangerous!	367
<code>\RequirePackageWithOptions</code> :		<code>\@xfloat</code> : Added <code>\@nodocument</code> to	
Reset <code>\@unprocessedoptions</code>		trap floats in the preamble	358
for /2269	462		

- 1996/10/24 ltoutput.dtx v1.1z
`\@addtocurcol`: Added `\nobreak`,
 etc as appropriate 421, 425
`\@specialoutput`: Added
`\nobreak` as appropriate . . . 402
`\@topnewpage`: Added
`\@nodocument` to trap
`\twocolumn` in the preamble 398
`\newpage`: Better checks for
 noskipsec and inlabel added,
 plus nobreak 397
- 1996/10/25 ltlists.dtx v1.0n
`\endtrivlist`: Change `\indent` to
`\leavevmode` 278
 Reset flags explicitly 278
- 1996/10/25 ltoutput.dtx v1.2a
`\newpage`: Reset all flags
 explicitly 397
- 1996/10/26 ltlists.dtx v1.0o
`\endtrivlist`: Correct typo 278
- 1996/10/27 ltoutenc.dtx v1.8c
`\@strip@args`: Removed macro . . 98
 General: Added `\r A` 106
 Added
`\textasteriskcentered` 103, 112
 Corrected syntax descriptions . . 92
 Removed `\aa` and `\AA` 102, 106, 108
- 1996/10/28 ltplain.dtx v1.1u
 General: (CAR) More doc changes 14
`\dotfill`: Removed math mode . . 29
- 1996/10/29 ltplain.dtx v1.1v
`\dotfill`: Got arithmetic correct
 (CAR) 29
- 1996/10/29 ltspaced.dtx v1.2u
`\@gnewline`: Added macro 70
`\@no@lnbk`: Macro replaces `\@lnbk`
 and `\@no@lnbk` 69
`\@`: Corrected and rationalised code 70
`\nolinebreak`: Reimplemented
 both using `\@no@lnbk` 69
- 1996/10/31 ltfinal.dtx v1.0z
 General: Added extra `\lcode`,
 hoping it does no harm in T1
 (pr/1969) 500, 504
- 1996/10/31 ltlists.dtx v1.0p
`\@trivlist`: Added check for
 missing item in outer list . . . 277
- 1996/10/31 ltsect.dtx v1.0y
 General: Corrected and tidied
 documentation; removed long
 lines 344
- 1996/11/03 ltplain.dtx v1.1w
`\dotfill`: Saved tokens by using
`\hb@xt@` 29
- 1996/11/04 lterror.dtx v1.2m
`\@nodocument`: Always define
`\@nodocument` in kernel, so that
 it can be cleared by `\document`. 61
- 1996/11/04 ltlists.dtx v1.0q
`\@trivlist`: Moved check for
 missing item: only checked
 when not inlabel flag is false 277
- 1996/11/05 ltfiles.dtx v1.1i
`\nofiles`: Standard `\if@nobreak`
 test added 85
- 1996/11/09 ltmath.dtx v1.1c
`\@ensuredmath`: Made long, as it
 was before. /2104 266
- 1996/11/18 ltfssbas.dtx v3.0s
`\define@newfont`: (DPC)
 lowercase fd file names.
 internal/1044 150
- 1996/11/18 ltoutenc.dtx v1.8d
 General: (DPC) lowercase external
 file names. internal/1044 . . . 119
- 1996/11/20 fontdef.dtx v2.2p
 General: lowercase fd and enc.def
 file names /1044 214
- 1996/11/20 ltvers.dtx v1.0f
 General: Check for old format
 modified /2319 32
- 1996/11/23 ltoutenc.dtx v1.8e
 General: Corrected description . . 92
 Extended description 93
- 1996/11/28 ltvers.dtx v1.0g
 General: Check for old format
 modified /2319 32
- 1996/12/06 ltdirchk.dtx v1.0u
`\IfFileExists`: *** removed from
 various messages for GNU
 Make. internal/2338 10
- 1996/12/06 ltfloat.dtx v1.1r
`\@caption`: Call `\@setminpage` if
 needed. latex/2318 357
- 1996/12/06 ltfssini.dtx v3.0h
 General: (DPC) Remove *** from
 messages internal/2338 212
- 1996/12/17 ltclass.dtx v1.0w
`\g@addto@macro`: Use `\begingroup`
 to save making a mathord . . 466
- 1996/12/20 ltsect.dtx v1.0z
`\@dottedtocline`: Added
`\nobreak` for latex/2343 . . . 353
- 1997/01/08 fontdef.dtx v2.2q
 General: Use
`\DeclareMathDelimiter` to set
 delimiter codes 222

<code>\mathparagraph</code> : Define using <code>\DeclareMathSymbol</code>	229	1997/05/07 ltspace.dtx v1.2v <code>\newline</code> : Made completely robust.	70
1997/01/08 ltfiles.dtx v1.1j <code>\@include</code> : reset <code>\deadcycles</code> latex/2365	86	1997/05/29 ltfstrc.dtx v3.0j General: Replaced <code>\</code> by <code>\MessageBreak</code> , as suggested by Donald Arseneau.	161
1997/01/08 ltmath.dtx v1.1d <code>\root</code> : (DPC) Remove spurious space tokens from plain \TeX definition /2359	260	1997/05/29 ltlogos.dtx v1.1f <code>\LaTeXe</code> : Added <code>\m@th</code> so that the $\LaTeX 2_{\epsilon}$ logo works with non-zero values of <code>\mathsurround</code>	80
1997/02/05 ltclass.dtx v1.0x <code>\g@addto@macro</code> : missing percent /2402	466	1997/06/16 ltdirchk.dtx v1.0v General: documentation improvements mainly from internal/2520	1
1997/02/21 ltlists.dtx v1.0r <code>\@item</code> : <code>\ifvoid</code> check added for <code>\noindent</code> . latex/2414	280	1997/06/16 ltfloat.dtx v1.1s General: documentation fixes . . .	354
1997/03/21 ltcounts.dtx v1.1e <code>\fnsymbol</code> : Use <code>\mathsection</code> and <code>\mathparagraph</code> . latex/2445	134	1997/06/16 ltfncmd.dtx v3.3v General: Fix typo in documentation.	236
1997/04/14 ltfiles.dtx v1.1k <code>\document</code> : Set the document space factor defaults. latex/2404	84	1997/08/05 ltoutenc.dtx v1.9e General: Corrected order of arguments in <code>\UseTextSymbol</code> example.	92
<code>\normalsfcodes</code> : Macro added (from patch file) latex/2404	85	1997/08/29 ltoutenc.dtx v1.9f General: Added OT4 encoding, provided by Marcin Woliński.	91
1997/04/14 ltoutput.dtx v1.2b <code>\@writsetup</code> : Call <code>\normalsfcodes</code> (from patch file) latex/2404	411	1997/09/09 ltdefns.dtx v1.2z <code>\providecommand</code> : Use <code>\begingroup</code> to avoid generating math ords if used in math mode. pr/2573	41
Move <code>\label</code> and <code>\index</code> (from patch file)	411	1997/09/15 ltpictur.dtx v1.1g <code>\@getcirc</code> : Warn if lines become invisible pr/2524	334
1997/04/24 ltbibl.dtx v1.1m <code>\@citex</code> : <code>\@empty</code> to avoid primitive error on empty cite keys. latex/2432	377	<code>\@picture@warn</code> : Macro added pr/2524	334
1997/04/30 ltoutenc.dtx v1.9a General: Changed <code>\textsc</code> to <code>\scshape</code>	104	<code>\@sline</code> : Warn if lines become invisible pr/2524	325
Introduced <code>\textcopyright</code> and modified <code>\copyright</code>	103	1997/10/06 ltcounts.dtx v1.1f <code>\@Roman</code> : Change <code>\@Roman</code> to be fully expandable, so that the result is written properly to files.	135
Introduced <code>\textcopyright</code> and modify <code>\copyright</code>	104	<code>\@slowromancap</code> : Macro added.	135
Modified <code>\textunderscore</code> , removing <code>\mathunderscore</code>	103	1997/10/08 ltlogos.dtx v1.1h <code>\LaTeX</code> : Simplify macro (force loading of suitable math fonts once).	80
Modified <code>\underscore</code> , removing <code>\mathunderscore</code>	104	1997/10/10 ltclass.dtx v1.0y <code>\endfilecontents</code> : <code>\currentenv</code> in banner	468
1997/04/30 ltoutenc.dtx v1.9b General: Added <code>\leavevmode</code> to <code>\textunderscore</code>	103		
1997/05/04 ltoutenc.dtx v1.9c General: Added ‘hex index tabs’	109		
Added TS1 encoding v2.2.beta	115		
1997/05/07 ltoutenc.dtx v1.9d General: Added <code>\leavevmode</code> to <code>\textcompwordmark</code>	103		

- `\reserved@c` not `\verbatim@out` to save a `cname` 468
- Check for text before or after `\end environment. latex/2636` 468
- Use `\@gobbletwo` 468
- 1997/10/17 `lftncmd.dtx v3.3w`
 - `\check@nocorr@`: Check for vertical mode moved here, from `\DeclareTextFontCommand` (see `PR/2646`). 240
 - `\DeclareTextFontCommand`: Reinstalled `\check@icr` as check is now done in `\check@nocorr@` (see `PR/2646`). 238
- 1997/10/20 `lftfinal.dtx v1.1a`
 - `\@uclclist`: Removed `\aa` and `\AA` from `\uclclist` as these are macros. 504
- 1997/10/21 `ltdfns.dtx v1.2z1`
 - `\renew@command`: Use `\begingroup/\endgroup` rather than braces for grouping, to avoid generating empty math atom. 39
- 1997/10/21 `lftssbas.dtx v3.0t`
 - `\define@newfont`: Move `\makeatletter` to `\nfss@catcodes`. 150
 - `\nfss@catcodes`: Moved `\makeatletter` from `\try@load@font@shape`. . . . 151
- 1997/11/09 `ltoutput.dtx v1.2c`
 - `\@specialoutput`: Remove incorrect code: only one `\@emptycol` is needed here . . 400
 - `\@topnewpage`: Documentation of `vsize` check enhanced 398
- 1997/11/13 `lftssdcl.dtx v3.0f`
 - `\DeclareSymbolFont`: (DPC) Really update `\group@list` dont leave new version in `\toks@. latex/2661` 193
 - `\stepcounter`: (DPC) Remove as never used. (Re)defined in `lccounts` 187
- 1997/11/19 `lfloat.dtx v1.1t`
 - `\@footnotetext`: Missing percent, again 372
- 1997/11/19 `ltoutput.dtx v1.2d`
 - `\@vtryfc`: Reindent code, to be understandable(DPC). 415
- 1997/11/20 `lftssdcl.dtx v3.0g`
 - `\document@select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a `cname` per math version as no reset list) 189
 - `\select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a `cname` per math version as no reset list) 187
- 1997/11/23 `ltoutenc.dtx v1.9g`
 - General: Use `\textperthousand`, `\textpertenthousand` and `\textfractionsolidus` not `\textpermill`, `\textpertenmill` and `\textfraction. /2673` 115
- 1997/12/17 `ltoutenc.dtx v1.9h`
 - General: Added `\textperthousand` and `\textpertenthousand` . . 107, 108
 - Added code for `textcomp.sty`. . 119
 - Added section. 119
 - Added `textcomp.sty`. 91
 - As in OT1, Added `\leavevmode` at start of `\c`, otherwise the output routine might be invoked within the macro. . . 107
 - Changed to decimal codes in `\localign`. 117
 - Changed to decimal codes. . . . 113
 - Documentation changes and additions. 91
 - Example corrected, braces removed. 91
 - Removed default settings, see next section. 115
- 1997/12/19 `ltoutenc.dtx v1.9i`
 - General: Documentation corrections. 91
- 1997/12/20 `fontdef.dtx v2.2s`
 - General: Added documentation . 216
- 1997/12/31 `ltoutenc.dtx v1.9k`
 - General: Further correction 92
- 1998/01/12 `ltoutenc.dtx v1.9k`
 - General: Added `\ProvidesPackage` for `textcomp.sty` 91
 - Adding missing braces and `\ushape`. 117
- 1998/01/16 `ltoutenc.dtx v1.9m`
 - General: fixed decimal codes. `latex/2734` 113
- 1998/03/04 `ltdfns.dtx v1.2z2`
 - `\@xargdef`: Unnecessary `\expandafter` removed: `pr/2758` 38

1998/03/05 ltoutenc.dtx v1.9n	1998/05/20 ltfinal.dtx v1.1b
General: Added masc/fem ords as in pr/2579	General: Set up lccodes before loading hyphenation files: pr/2639
1998/03/20 ltdefns.dtx v1.2z3	Set up uc/lccodes after loading hyphenation files: pr/2639 . .
\@thirdofthree: Macro added . .	1998/05/28 lterror.dtx v1.2n
1998/03/20 ltoutenc.dtx v1.9o	\@notdefinable: Added message re ‘end...’ pr/1555
General: Added various	1998/06/04 ltboxes.dtx v1.1c
\UndeclareTextCommand	\@rule: Support calc-expressions
declarations for pr/2783 . . .	1998/06/12 ltoutenc.dtx v1.9p
Documentation added about	General: Corrected 130 and 131, see pr/2834
order of decls	Renamed \textmacron pr/2840
Documentation added for	1998/06/12 ltoutenc.dtx v1.9q
pr/2783	\add@accent: Explicitly set \spacefactor after \accent (pr/2877)
Load decls after defaults for speed.	1998/06/18 lttab.dtx v1.1k
\UndeclareTextCommand: Macro added for pr/2783	General: Small addition to documentation
1998/03/21 ltclass.dtx v1.0z	1998/07/06 lttab.dtx v1.1l
General: Added to documentation of filecontents	General: Small correction to documentation
1998/03/21 ltclass.dtx v1.1a	1998/08/17 ltboxes.dtx v1.1e
\@providesfile: Allow &. Internal/2702	General: (RmS) Minor Documentation fixes.
General: Correct to new onlypreamble command list .	1998/08/17 ltclass.dtx v1.1c
1998/03/25 ltssbas.dtx v3.0u	General: (RmS) Minor documentation fixes.
\showhyphens: Suppress unnecessary error when used in preamble	1998/08/17 ltdirchk.dtx v1.0w
1998/04/11 fontdef.dtx v2.2t	General: (RmS) Documentation improvements.
General: Added \mathring accent (pr2785)	1998/08/17 ltfntcmd.dtx v3.3x
1998/04/15 fontdef.dtx v2.2u	General: (RmS) Minor documentation fixes.
General: Use new syntax for \DeclareMathDelimiter . . .	1998/08/17 ltssbas.dtx v3.0v
1998/04/15 ltssdcl.dtx v3.0h	General: (RmS) Documentation fixes.
\@xxDeclareMathDelimiter: Macro added (pr/2662)	1998/08/17 ltfssdcl.dtx v3.0i
1998/04/17 fontdef.dtx v2.2v	General: (RmS) Corrected minor glitches in changes entries. . .
General: Reinsert symbol defs for < and > chars.	1998/08/17 ltfssini.dtx v3.0i
1998/04/18 fontdef.dtx v2.2w	General: (RmS) Minor documentation fixes.
General: Reinsert symbol def for / char.	1998/08/17 ltlogos.dtx v1.1i
1998/05/07 ltclass.dtx v1.1b	General: (RmS) Minor documentation fixes.
\@fileswithoptions: Modify help message for latex/2805	1998/08/17 ltmath.dtx v1.1c
1998/05/18 lttab.dtx v1.1j	General: (RmS) Minor documentation fixes.
\@endpbox: Use \setlength to set \hspace, so that the changes in the calc package apply here. . .	General: (RmS) Minor documentation fixes.
\@tabular*: Use \setlength, so that calc extensions apply. . .	

1998/08/17 ltmiscen.dtx v1.1g	1999/01/18 ltdefs.dtx v1.3c
General: (RmS) Minor	<code>\@yargd@f</code> : New implementation
documentation fixes. 249	DPC /2942 38
1998/08/17 ltspac.dtx v1.2w	1999/02/09 ltdefs.dtx v1.3d
General: Documentation fixes. . . 66	<code>\@yargd@f</code> : catch bad argument
1998/08/17 preload.dtx v2.1g	forms by re-inserting #3 38
General: (RmS) Minor	1999/02/12 ltssini.dtx v3.0j
documentation fixes. 232	<code>\oldstylenums</code> : Use <code>\rmdefault</code>
1998/09/19 ltoutenc.dtx v1.9r	instead of <code>cmm</code> (pr/2954) 210
<code>\a</code> : Added <code>\string</code> (pr/2878) . . . 102	1999/02/24 ltoutenc.dtx v1.9t
1998/11/13 lttab.dtx v1.1m	General: Corrected hackery cyrillic
<code>\@array</code> : Check for hmode to see if	uc/lc list 119
something went wrong during	1999/03/01 ltdefs.dtx v1.3e
parsing (pr/2884) 308	<code>\@ifnextchar</code> : remove extra
1999/01/05 fontdef.dtx v2.2x	<code>\long</code> . internal/2967 46
General: Need special protection	1999/04/15 ltpictur.dtx v1.1h
for character > in <code>\changes</code>	<code>\@getlarrow</code> : Replaced octal
entry. 214	number, CAR 326
1999/01/06 ltssbas.dtx v3.0w	<code>\@upvector</code> : Replaced octal
<code>\DeclareFontEncoding</code> : Added	number, CAR 327
<code>\LastDeclaredEncoding</code> to	General: Replaced octal number,
support cyrillic integration	CAR 326, 327
(pr/2988) 142	Replaced octal numbers, CAR 318
<code>\LastDeclaredEncoding</code> : Added	1999/04/19 ltfloat.dtx v1.1u
<code>\LastDeclaredEncoding</code> to	<code>\caption</code> : Made caption an error
support cyrillic integration	outside a float: latex/2815 . . 357
(pr/2988) 142	1999/04/27 ltboxes.dtx v1.1f
1999/01/06 ltoutenc.dtx v1.9r	<code>\@parboxto</code> : (CAR) Changed
<code>\@strip@args</code> : New impl for	<code>\@empty</code> to <code>\relax</code> as flag for
latex/2930 99	natural width: pr/2975 291
General: Minor documentation	1999/04/29 ltdefs.dtx v1.3f
fix. 117	<code>\@yargd@f</code> : Full expansion and
1999/01/06 ltoutput.dtx v1.2e	conversion needed for digit in
<code>\@makecol</code> : Added negative vskip,	new version, see pr/3013 38
as when processing outputbox	New macro added 38
below: suggested by Fred	1999/06/10 ltoutenc.dtx v1.9u
Bartlett pr/2892 406	General: Ensure that we also
1999/01/07 ltdefs.dtx v1.3a	forget old options (pr/2888) . 121
<code>\@ifnextchar</code> : made long 46	1999/06/12 ltoutenc.dtx v1.9v
<code>\@newenvb</code> : made long and brace	General: Extend <code>\@uclclist</code> only
optional arg. latex/2896 40	once 120
<code>\@testopt</code> : made long and brace	1999/10/09 ltmath.dtx v1.1e
optional arg. latex/2896 38	<code>\active@math@prime</code> : Macro
1999/01/07 ltdefs.dtx v1.3b	added, see PR 3104. 262
<code>\@ifnextchar</code> : extra <code>\long</code> .	<code>\prime@s</code> : Introduce
latex/2902 46	<code>\active@math@prime</code> 262
1999/01/07 ltoutenc.dtx v1.9r	1999/10/09 ltoutput.dtx 1.2f
General: Hackery to allow using	<code>\@activechar@info</code> : Reset
fontenc several times 121	definition of active prime
Hackery to temp support cyrillic	character (used in math
uc/lc 119	mode) 409
1999/01/13 ltoutenc.dtx v1.9s	1999/10/28 ltoutenc.dtx v1.9w
<code>\@strip@args</code> : Simplified solution	<code>\add@accent</code> : Give
for latex/2930 99	<code>\accent@spacefactor a</code>

- default definition (pr/3084) . . . 98
- 1999/12/08 ltoutenc.dtx v1.9x
 General: Changed `\CYRRHOOK` and
`\cyrrhook` to `\CYRRHK` and
`\cyrrhk` as name changed in
 the cyrillic bundle for naming
 consistency with other “hook”
 glyphs. 119
- 2000/01/07 ltmiscen.dtx v1.1h
`\@verbatim`: Disable hyphenation
 even if the font allows it. . . . 255
- 2000/01/15 ltpictur.dtx v1.1i
`\@upvector`: Removed space at
 end-of-line, CAR 327
- 2000/01/30 ltfntcmd.dtx v3.3y
`\DeclareTextFontCommand`: Use
`\hmode@bgroup` now (pr/3160) 238
- 2000/01/30 ltoutenc.dtx v1.9y
 General: Use `\hmode@bgroup`
 where applicable
 (pr/3160) 105–107, 112–115, 117
`\add@accent`: Use `\hmode@bgroup`
 where applicable (pr/3160) . . . 97
`\hmode@bgroup`: Macro added . . . 98
- 2000/01/30 ltoutenc.dtx v1.9z
`\@use@text@encoding`: Macro
 reimplemented (pr/3160) 99, 100
`\add@accent`: Macro
 reimplemented (pr/3160) 97
`\hmode@start@before@group`:
 Macro added (pr/3160) 100
- 2000/05/19 ltmiscen.dtx v1.1i
`\enddocument`: Reset
`\AtEndDocument` for
 latex/3060 250
- 2000/05/26 ltpage.dtx v1.0j
`\@markright`: Reimplementation to
 fix expansion error (pr/3203). 381
`\leftmark`: Use `\@empty` instead of
 brace group (pr/3203). 381
`\markright`: Reimplementation to
 fix expansion error (pr/3203). 381
`\rightmark`: Use `\@empty` instead
 of brace group (pr/3203). . . . 381
- 2000/06/02 ltpage.dtx v1.0k
`\@markright`: Small adjustment to
 give slightly less expansion,
 CAR 381
`\markright`: Small adjustment to
 give slightly less expansion,
 CAR 381
 Tidied 1.0j reimplementation,
 CAR 381
- 2000/07/11 ltmiscen.dtx v1.1j
`\enddocument`: Fix typo in
 warning 251
- 2000/07/12 ltoutput.dtx 1.2g
 General: Ensure that rule is in
`\normalcolor` 447
- 2000/07/12 ltoutput.dtx 1.2i
`\@makecol`: Removed negative
 vskip, as it gives unacceptable
 results when the depth is large:
 pr/3189 406
- 2000/07/19 ltoutput.dtx v1.2h
`\@writsetup`: Reset and restore
`\@if@newlist` for
 internal/3231 410
- 2000/08/30 ltoutenc.dtx v1.91
`\@use@text@encoding`: Rearranged
 but no change to final code,
 CAR (pr/3160) 99
`\add@accent`: Rearranged but no
 change to final code, CAR
 (pr/3160) 97
- 2000/09/01 ltfinal.dtx v1.1d
`\errhelp`: Set error help empty at
 very end (pr/449 done
 correctly). 506
- 2000/09/24 ltfloat.dtx v1.2b
`\end@dblfloat`: FMI: use output
 routine to defer float 362
- 2000/09/24 ltoutput.dtx v1.2b
`\@docclearpage`: FMI: ensure
`\docclearpage` is called again
 until all floats are output. . . . 404
- 2000/09/24 ltoutput.dtx v1.2n
`\@addtocurcol`: FMI: test for wide
 float was in wrong place 420
- 2001/01/07 ltoutput.dtx v1.2j
`\@writsetup`: And do it in the
 right macro (pr/3286) 410
- 2001/02/16 ltxref.dtx v1.1k
`\@newl@bel`: Added an extra
 grouplevel (PR3250), jlb 246
- 2001/05/25 ltclass.dtx v1.1d
`\@providesfile`: Explicitly set
 catcode of `\endlinechar` to 10
 (pr/3334) 459
- 2001/05/25 ltdirchk.dtx v1.0x
 General: Explicitly set catcode of
`\endlinechar` to 10 (pr/3334) . . 4
- 2001/05/28 ltoutenc.dtx v1.93
 General: Added composites for
 compatibility with T1,
 pr/3295 107

Changed the effect of <code>\.i</code> , pr/3295	109	2002/10/01 ltfloat.dtx v1.1v <code>\themppfootnote</code> : Use braces around <code>\itshape</code> to keep font change local (pr/3460).	370
2001/06/02 fontdef.dtx v2.2y General: Provide default cfg files (pr/3264)	230	2002/10/02 ltffssbas.dtx v3.0x <code>\DeclareFontSubstitution</code> : Adding <code>\LastDeclaredEncoding</code> introduced a bug as on some occasions that macro name was stored in the internal lists instead of the actual encoding. (pr/3459)	142
2001/06/04 fontdef.dtx v2.2z General: Guard against math active equal and pipe sign in <code>\models</code> (pr/3333)	226	2002/10/28 ltlists.dtx v1.0s <code>\endtrivlist</code> : Check for math mode (pr/3437)	278
Guard against math active equal sign in <code>\Relbar</code> (pr/3333)	226	2002/10/28 ltoutenc.dtx v1.96 General: coding change, to follow bug fix by DEK in plain.tex (pr/3469)	106, 114
2001/06/04 ltclass.dtx v1.1e <code>\@providesfile</code> : But only if it is a char (pr/3334)	459	2002/12/13 ltbibl.dtx v1.1n <code>\@citex</code> : Added <code>\leavevmode</code> in case citation is at start of paragraph (pr/3486)	377
2001/06/04 ltdirchk.dtx v1.0y General: But only if it is a char (pr/3334)	4	2003/01/01 ltfntcmd.dtx v3.3z General: Code checked and documentation extended by Chris	238
2001/06/04 ltpictur.dtx v1.1j <code>\@sline</code> : Don't warn for exactly zero pr/3318	325	2003/05/18 ltbibl.dtx v1.1o <code>\nocite</code> : Check if we are after <code>\document</code>	378
2001/06/04 ltvers.dtx v1.0i General: Check for old format disabled	32	2003/08/27 ltpictur.dtx v1.1k <code>\@bezier</code> : added missing displacement pr/3566	339
2001/06/05 ltoutenc.dtx v1.94 General: Text composite Commands need kludges for ‘,’ – see tlb1903.lvt	107	<code>\@sline</code> : check for <code>\@linechar</code> being empty pr/3570	325
2001/08/26 ltclass.dtx v1.1f <code>\@providesfile</code> : Readded setting of space char (pr/3353)	459	2003/10/13 ltfinal.dtx v1.1e General: Added extra <code>\lccode</code> for <code>\-</code> and <code>\textcompwordmark</code>	500
2002/02/24 ltplain.dtx v1.1x <code>\loggingall</code> : Macro added	29	2003/12/16 ltoutput.dtx v1.2k <code>\@makecol</code> : Ensure that <code>\@elt</code> has a defined state (pr/3586)	406
<code>\loggingoutput</code> : Macro added	29	2003/12/30 ltpictur.dtx v1.1j <code>\@getcirc</code> : issue warning if circle size can't be met pr/3473	334
<code>\showoutput</code> : Use newly added <code>\loggingoutput</code>	29	2004/01/03 ltoutenc.dtx v1.99b General: Added <code>\textogonekcentered</code> (pr/3532)	107
<code>\tracingall</code> : Use newly added <code>\loggingoutput</code>	29	Added composites for <code>\k</code> (pr/3532)	112
2002/06/16 ltoutenc.dtx v1.95 General: Added <code>\textbardbl</code> (pr/3400)	112	Use <code>\ooalign</code> for <code>\k</code> (pr/3532)	107
Added default for <code>\textbardbl</code> (pr/3400)	103	2004/01/04 ltbibl.dtx v1.1p <code>\nocite</code> : Changed error message	378
2002/06/17 ltoutenc.dtx v1.95 General: Corrected <code>\c</code> for T1 (pr/3442)	107		
Definition of <code>\textexclamdown</code> changed (pr/3368)	106		
Definition of <code>\textquestiondown</code> changed (pr/3368)	106		
2002/06/18 ltoutenc.dtx v1.95 General: Changed def for <code>\textregistered</code> to avoid small caps (pr/3420)	104		

2004/01/04 ltoutenc.dtx v1.99c		<code>\DeclareSymbolFontAlphabet:</code>	
General: More adjustments for		(MH) Make document	
ogonek (pr/3532)	107	commands robust	205
2004/01/23 ltdefns.dtx v1.1g		<code>\new@mathalphabet:</code> (MH) Make	
<code>\@newenva:</code> Use kernel version of		document commands robust	196
<code>\@ifnextchar</code> (pr/3501)	40	<code>\non@alpherr:</code> (MH) Change	
<code>\@testopt:</code> Use kernel version of		because command is now	
<code>\@ifnextchar</code> (pr/3501)	38	properly robust	189
<code>\@xargdef:</code> Use kernel version of		<code>\SetMathAlphabet:</code> (MH) Make	
<code>\@ifnextchar</code> (pr/3501)	37	document commands robust	197
<code>\@xdblarg:</code> Use kernel version of		2005/09/27 ltoutenc.dtx v1.99g	
<code>\@ifnextchar</code> (pr/3501)	47	General: Replace <code>\sh@ft</code> by	
2004/01/23 ltdefns.dtx v1.3g		<code>\ltx@sh@ft</code>	105, 107, 113
<code>\kernel@ifnextchar:</code> Added		2005/09/27 ltplain.dtx v1.1y	
macro (pr/3501)	46	<code>\ltx@sh@ft:</code> New macro	29
2004/01/28 ltclass.dtx v1.1g		<code>\sh@ft:</code> Macro no longer used but	
<code>\@providesfile:</code> Use kernel		left for compatibility	28
version of <code>\@ifnextchar</code>		2005/11/08 ltoutenc.dtx v1.99h	
(pr/3501)	459	General: Added <code>\ij</code> and <code>\IJ</code> from	
2004/01/28 ltvers.dtx v1.0k		babel. (pr/3771)	103, 106, 108
General: Check for old format		2005/11/10 ltmath.dtx v1.1g	
made 5 years (pr/3601)	32	<code>\l:</code> (MH) Fixed potential problem	
2004/02/02 fontdef.dtx v2.3		in <code>\l</code> (pr/3399).	263
General: Many things from here on		General: (MH) Minor	
made robust	226	documentation fixes.	258
2004/02/04 fontdef.dtx v2.3a		2006/05/18 ltboxes.dtx v1.1g	
General: Added bigtriangle		<code>\@parboxto:</code> Ensure <code>\@parboxto</code>	
synonyms for <code>stmaryrd</code>	224	holds the value of <code>\@tempdimb</code>	
2004/02/04 ltspc.dtx v1.3		not the register itself	
<code>\nobreakdashes:</code> (Macro added	77	(pr/3867)	291
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>	448
pr/3563	95	2007/08/05 ltclass.dtx v1.1h	
2004/02/07 ltoutput.dtx v1.2l		<code>\@fileswithoptions:</code> Prevent loss	
<code>\@docclearpage:</code> Empty kludgeins		of brackets PR/3965	464
box if necessary, pr/3528	404	2007/08/06 ltctrl.dtx v1.0h	
2004/02/13 ltoutenc.dtx v1.99e		<code>\@fornoop:</code> Really make defs long	53
General: Documentation fixes:		2007/08/31 ltssdcl.dtx v3.0l	
typos	91	<code>\SetSymbolFont@:</code> Font warning	
2004/02/15 ltbibl.dtx v1.1q		changed to info for encoding	
<code>\@cite@ofmt:</code> Added hook with		change (pr/3975)	194
default value <code>\hbox</code>	379	2009/09/24 ltvers.dtx v1.0l	
<code>\@citex:</code> Changed to use a hook		General: Stop checking for old	
with default value <code>\hbox</code>	378	format	32
2004/02/15 ltspc.dtx v1.3a		2009/10/20 ltssdcl.dtx v3.0m	
<code>\nobreakdashes:</code> (Added		<code>\in@:</code> More robust thanks to	
spacefactor setting	77	Heiko.	185
2004/10/20 ltoutput.dtx v1.2m		2009/10/28 ltoutenc.dtx v1.99k	
<code>\@makecol:</code> Removed dead code	406	General: Added Latin Modern and	
2005/07/27 ltssdcl.dtx v3.0j		TeX Gyre subsets	130
<code>\DeclareMathAlphabet:</code> (MH)		2009/11/04 ltoutenc.dtx v1.99l	
Make document commands		General: Added more Latin	
robust	195	Modern and TeX Gyre subsets	130

2009/12/14 ltfntcmd.dtx v3.4a	<code>\textsubscript</code> : Command added (latexrelease)	371
<code>\ifmaybe@ic</code> : Macro added	240	
<code>\maybe@ic@</code> : Use switch		
<code>\ifmaybe@ic</code> instead of		
<code>\if@tempswa</code>	240	
<code>\t@st@ic</code> : Use switch <code>\ifmaybe@ic</code>		
instead of <code>\if@tempswa</code>	241	
2010/08/17 ltmiscen.dtx v1.1k		
<code>\enddocument</code> : Use braces around		
<code>\input</code> arg (pr/4124)	251	
2010/08/17 ltmiscen.dtx v1.1l		
<code>\enddocument</code> : Change of plan: use		
<code>\@@input</code> instead (pr/4124)	251	
2011/05/08 ltfssdcl.dtx v3.0n		
<code>\in@</code> : Simplified thanks to Bruno.	185	
2011/08/19 ltclass.dtx v1.1i		
<code>\@ifclasswith</code> : Re-jig definition		
after more stringent <code>\in@</code> test.	458	
2011/09/03 ltfssdcl.dtx v3.0o		
<code>\new@mathversion</code> : (Will) Remove		
<code>\global</code> before <code>\newcount</code>		
(unnecessary and caused etex		
bug).	192	
2012/01/20 ltplain.dtx v2.0b		
<code>\loggingall</code> : etex tracing if		
available	29	
2013/07/07 ltclass.dtx v1.1i		
General: Correctly describe how		
the date in <code>\@ifpackagelater</code>		
is used	455	
2014/04/18 ltoutput.dtx v1.1o		
General: Handle infinite glue from		
<code>\enlargethispage</code> (pr/4023)	447	
2014/04/24 ltoutput.dtx v1.2n		
<code>\fl@tracemessage</code> : Renamed		
internal trace commands;		
provide as package	438	
2014/04/27 ltfloat.dtx v1.2b		
<code>\end@dblfloat</code> : Inline the code to		
allow some coexistence with		
packages that hook into		
<code>\end@float</code> and do not know		
about the algorithm change	362	
2014/06/10 ltfloat.dtx v1.2b		
<code>\end@dblfloat</code> : missing <code>\fi</code>		
added	362	
2014/12/30 ltfinal.dtx v2.0a		
<code>\newmarks</code> : macro added	496	
<code>\newXeTeXintercharclass</code> : macro		
added	496	
2014/12/30 ltfloat.dtx v1.2a		
<code>\textsubscript</code> : Command		
added (latexrelease)	371	
2014/12/30 ltfssbas.dtx v3.0y		
<code>\mathgroup</code> : move allocation to		
ltplain.	139	
2014/12/30 ltoutput.dtx v1.2m		
General: Command updated		
(latexrelease)	447	
2014/12/30 ltplain.dtx v2.0a		
<code>\e@alloc</code> : macro added	19	
<code>\e@alloc@chardef</code> : macro added	18	
<code>\e@alloc@top</code> : macro added	18	
<code>\e@ch@ck</code> : macro added	19	
<code>\extrafloats</code> : macro added	19	
<code>\newlanguage</code> : New engine-specific		
allocation scheme		
(latexrelease)	17	
2014/12/30 ltspcex.dtx v1.3b		
<code>\@</code> : <code>\@</code> discards spaces when		
moving (pr3039)(latexrelease)	78	
2015/01/03 ltdefns.dtx v1.4a		
<code>\typein</code> : use modified definition in		
luatex	36	
2015/01/03 ltdirchk.dtx v1.1		
General: Enable extra primitives		
when Lua _{TeX} is used	3	
2015/01/03 ltfinal.dtx v2.0a		
General: Skip resetting codes with		
Unicode engines	503	
Unicode data loading added	498	
2015/01/07 ltvers.dtx v1.0n		
<code>\IncludeInRelease</code> : macro added	33	
2015/01/08 ltboxes.dtx v1.1h		
<code>\framebox</code> : Make Robust		
(latexrelease)	289	
<code>\makebox</code> : Make Robust		
(latexrelease)	285	
<code>\parbox</code> : Make Robust		
(latexrelease)	290	
<code>\raisebox</code> : Make Robust		
(latexrelease)	294	
<code>\rule</code> : Make Robust		
(latexrelease)	293	
<code>\savebox</code> : Make Robust		
(latexrelease)	287	
2015/01/08 ltdefns.dtx v1.4a		
<code>\MakeRobust</code> : Added macro	45	
2015/01/08 ltlength.dtx v1.1c		
<code>\setlength</code> : to ensure first length		
argument is terminated.		
(latexrelease)	138	
2015/01/08 ltmath.dtx v1.1h		
<code>\)</code> : Make Robust (latexrelease)	262	
<code>\]</code> : Make Robust (latexrelease)	263	

2015/01/09 ltssini.dtx v3.1a	2015/01/14 ltsspace.dtx v1.3e
<code>\em</code> : Allow <code>\emph</code> to produce small caps (latexrelease)	<code>\addpenalty</code> : Avoid adding redundant skips (DPC)
209	75
<code>\emminnershape</code> : macro added (latexrelease)	2015/01/17 ltvers.dtx v1.0m
209	<code>\IncludeInRelease</code> : modified with <code>\@currname</code>
2015/01/09 ltsspace.dtx v1.1h	33
<code>\addpenalty</code> : Donald Arseneau's fix from PR/377703 (latexrelease)	2015/01/19 ltvers.dtx v1.0o
75	<code>\IncludeInRelease</code> : Optional argument
2015/01/10 ltcounts.dtx v1.1h	33
<code>\@fnssymbol</code> : Unse <code>\TextOrMath</code> (latexrelease)	2015/01/20 ltoutput.dtx v1.2m
135	<code>\fl@tracemessage</code> : Reset <code>\IncludeInRelease</code> flags
<code>\@stpelit</code> : Reset all within counters in one go (latexrelease)	439
133	2015/01/22 ltvers.dtx v1.0p
2015/01/11 ltcounts.dtx v1.1h	General: Preserve any <code>\everyjob</code> material inserted by a loader (.ini file)
<code>\TextOrMath</code> : Add command to solve robustness issues (pr/3752) (latexrelease)	33
136	2015/01/23 ltfinal.dtx v2.0b
2015/01/11 ltfloat.dtx v1.2b	<code>\newmarks</code> : use reserved count 256
<code>\@dblfloatplacement</code> : float order in 2-column (latexrelease)	<code>\newXeTeXintercharclass</code> : use reserved count 257
364	496
<code>\@xfloat</code> : Check for valid option (latexrelease)	2015/01/23 ltplain.dtx v2.0c
358	<code>\extrafloats</code> : reserve counts 256–265
<code>\end@dblfloat</code> : float order in 2-column (latexrelease)	19
362	2015/01/24 ltfinal.dtx v2.0c
2015/01/11 ltssbas.dtx v3.0y	General: Skip T1-code entirely with Unicode engines
<code>\@DeclareMathSizes</code> : Allow arbitrary units (latexrelease)	498
144	2015/02/03 ltfinal.dtx v2.0d
2015/01/11 ltsspace.dtx v1.3d	General: Set <code>\lccode</code> for – with Unicode engines
<code>\@Esphack</code> : Allow hyphenation (Donald Arseneau pr/3498) (latexrelease)	499
73	2015/02/16 ltoutenc.dtx v1.99m
<code>\@esphack</code> : Allow hyphenation (Donald Arseneau pr/3498) (latexrelease)	General: Added <code>\textcommabelow</code> latex/4414
72	104
2015/01/14 ltoutput.dtx v1.2n	Added <code>\lmtt</code> (Heiko Oberdiek) latex/4415
<code>\@addtocurcol</code> : float order in 2-column (latexrelease)	130
419	2015/02/16 ltoutenc.dtx v1.99n
<code>\@addtodblcol</code> : float order in 2-column (latexrelease)	General: Added <code>\textcommaabove</code>
430	112
<code>\@addtonextcol</code> : float order in 2-column (latexrelease)	Added composites for <code>\c</code>
426	107
<code>\@docclearpage</code> : Empty kludgeins box if necessary, pr/3528	2015/02/19 ltvers.dtx v1.0q
403	<code>\IncludeInRelease</code> : Swap argument order
float order in 2-column (latexrelease)	33
403	2015/02/20 ltplain.dtx v2.0d
<code>\@startdblcolumn</code> : float order in 2-column (latexrelease)	<code>\loggingall</code> : Spell commands correctly :-).
414	29
<code>\@xtryfc</code> : float order in 2-column (latexrelease)	2015/02/21 ltdefs.dtx v1.4b
415	General: Removed autoloading support
<code>\@ztryfc</code> : float order in 2-column (latexrelease)	34
417	2015/02/21 lterror.dtx v1.2o
	General: Removed autoloading support
	55
	2015/02/21 ltfiles.dtx v1.1m
	General: Removed autoloading support
	81

2015/02/21 ltssbas.dtx v3.0z		compatibility with older xelatex initialisation	496
General: Removed autoload code	139		
2015/02/21 ltsscmp.dtx v3.0d		2015/05/10 ltlists.dtx v1.0t	
General: Removed autoload code	181	\@doendpe: Explicitly reset \clubpenalty before clearing \everypar; see also pr/0462 and pr/4065	279
2015/02/21 ltssdcl.dtx v3.0p		2015/06/19 ltfinal.dtx v2.0g	
General: Removed autoload code	185	\newmarks: Use -1 for first range to get contiguous allocation .	496
2015/02/21 ltssstrc.dtx v3.0k		\Xe@alloc@intercharclass: Use -1 for first range to get contiguous allocation	496
General: Removed autoload code	159	2015/06/19 ltplain.dtx v2.0h	
2015/02/21 ltoutenc.dtx v1.99m		General: delete spurious old definition of \newtoks	22
General: Removed autoload code	91	\Xe@alloc: extra braces in case arguments not single token . .	19
2015/02/21 ltoutput.dtx v1.2n		\newlanguage: Use -1 for first range to get contiguous allocation	17
General: Removed autoload code	383	2015/06/23 ltfinal.dtx v2.0h	
\f@depth: macro added(latexrelease)	402	General: set \patch@level in ltvers rather than in ltfinal/ltpatch	505
2015/02/21 ltpictur.dtx v1.1k		2015/06/23 ltvers.dtx v1.0t	
General: Removed autoload code	318	General: set \patch@level in ltvers rather than in ltfinal/ltpatch	32
2015/02/21 ltplain.dtx v2.0e		2015/08/06 ltplain.dtx v2.0i	
General: Removed autoload code	14	\extrafloats: Add \string in case argument is not an unexpandable primitive	19
2015/02/21 lttab.dtx v1.1n		2015/08/23 ltdirchk.dtx v1.2	
General: Removed autoload code	296	General: Do not use luatex prefix .	3
2015/02/21 ltvers.dtx v1.0r		2015/08/23 ltvers.dtx v1.0v	
General: Removed autoload code	32	General: Allow negative patchlevel for pre-release	33
2015/02/22 ltsscmp.dtx v3.0e		2015/08/30 ltplain.dtx v2.1a	
General: Moved all code into latexrelease - obsolete commands are no longer automatically part of the kernel	181	\newinsert: new \newinsert implementation	21
2015/03/02 ltplain.dtx v2.0f		2015/09/24 ltluatex.dtx v1.0a	
\Xe@mathgroup@top: macro added	18	call_callback: Function added	492
\newlanguage: allow 255 math groups in Unicode engines . . .	17	callback.register: Function modified	490
2015/03/10 ltplain.dtx v2.0g		callback_descriptions: Function added	494
\hideoutput: macro added	30	\catcodetable@atletter: Macro added	479
\loggingall: Reorganise to be less noisy	29	\catcodetable@initex: Macro added	479
\tracingnone: macro added	30	\catcodetable@latex: Macro added	479
2015/03/18 ltssdcl.dtx v3.0q			
\DeclareSymbolFont: Restrict Symbol fonts to 0-15	193		
\document@select@group: Introduce \Xe@mathgroup@top	189		
\select@group: Introduce \Xe@mathgroup@top	187		
2015/03/26 ltfinal.dtx v2.0d			
General: Use renamed unicode-letters.def	498		
2015/04/07 ltssbas.dtx v3.1a			
\wrong@fontshape: Try loading fd file if family has changed . . .	152		
2015/04/28 ltfinal.dtx v2.0f			
\newXeTeXintercharclass: define \Xe@alloc@intercharclass for			

<code>\catcodetable@string</code> : Macro added	479	2015/11/27 ltluatex.dtx v1.0h	
<code>add_to_callback</code> : Function added	492	<code>callback_descriptions</code> : Match test in in-callback latex/4445	494
<code>remove_from_callback</code> : Function added	493	<code>in_callback</code> : Guard against undefined list latex/4445 . . .	494
<code>new_attribute</code> : Function added	486	2015/11/29 ltluatex.dtx v1.0i	
<code>disable_callback</code> : Function added	494	General: Declare this as local before used in the module error definitions (PHG)	483
<code>in_callback</code> : Function added . .	494	<code>call_callback</code> : Check name is not nil in error message (PHG)	492
<code>\newattribute</code> : Macro added . . .	479	<code>create_callback</code> : Check name is not nil in error message (PHG)	491
<code>\newcatcodetable</code> : Macro added	479	2015/12/02 ltluatex.dtx v1.0j	
<code>\newluabytecode</code> : Macro added .	481	General: Adjust hashtokens to store the result of <code>tex.hashtokens()</code> , not the function (PHG)	485
<code>\newluachunkname</code> : Macro added	482	Assorted typos fixed (PHG) . .	477
<code>\newluafunction</code> : Macro added .	481	Declaration/use of <code>first_head</code> fixed (PHG)	484
<code>\newwhatsit</code> : Macro added	481	Remove nonlocal iteration variables (PHG)	477
<code>module_error</code> : Function added .	485	Remove unreachable code after calls to <code>error()</code> (PHG)	477
<code>module_info</code> : Function added . .	485	2015/12/02 ltluatex.dtx v1.0k	
<code>module_warning</code> : Function added	485	General: resolve name and <code>i.description</code> (PHG)	490
<code>modules</code> : Function modified	483	<code>call_callback</code> : Give more specific error messages (PHG)	492
<code>create_callback</code> : Function added	491	<code>add_to_callback</code> : Give more specific error messages (PHG)	492
<code>provides_module</code> : Function added	483	<code>remove_from_callback</code> : adjust initialisation of <code>cb local</code> (PHG)	493
<code>luatexbase</code> : Table added	483	Give more specific error messages (PHG)	493
2015/10/02 ltdirchk.dtx v1.2a		<code>create_callback</code> : Give more specific error messages (PHG)	491
General: Allow backing out of unprefixed names	3	2015/12/10 ltfinal.dtx v2.0i	
2015/10/02 ltluatex.dtx v1.0e		General: Use new common Unicode data loaders	498
<code>uninstall</code> : Function added	495	2015/12/18 ltluatex.dtx v1.0l	
2015/10/03 ltluatex.dtx v1.0f		General: Load Unicode data from source	479
<code>provides_module</code> : use <code>luatexbase_log</code>	483	2016/01/04 ltfinal.dtx v2.0j	
2015/10/27 ltplain.dtx v2.1b		General: Do not set up inter character classes for XeTeX .	498
<code>\extrafloats</code> : Use global assignment when switching to extended range	19	<code>\xe@alloc@intercharclass</code> : Start allocation at one not three . .	496
2015/11/07 ltspacex.dtx v1.3f			
<code>\@esphack</code> : Only space if there is no space at the end of the hlist latex/4443	72		
2015/11/14 ltluatex.dtx v1.0g			
General: Track LuaTeX changes for <code>(new)token.create</code>	485		
2015/11/18 ltplain.dtx v2.2a			
<code>\newlanguage</code> : Extended stream allocation in <code>luatex (0.85)</code> . . .	17		
2015/11/19 ltplain.dtx v2.2b			
<code>\newlanguage</code> : Only extend allocation of write streams (see <code>luatex list</code>)	17		

2016/01/05 ltfinal.dtx v2.0k	token_filter removed	489
\Xe@alloc@intercharclass:	2016/02/18 ltfssdcl.dtx v3.0r	
Remove duplicated code . . .	\@DeclareMathDelimiter: Check	
2016/01/05 ltfinal.dtx v2.0l	for delimiter not \delimiter	202
General: Correct latexrelease	\DeclareMathAlphabet: Check for	
guards	mathaccent not \mathacemnt	199
Ensure old definitions for	\DeclareMathRadical: Check for	
inter-character class toks are	radical not \radical	204
available using latexrelease . .	\DeclareMathSymbol: Check for	
Missing brace	mathchar not \mathchar . . .	200
2016/01/05 ltfinal.dtx v2.0m	2016/03/13 ltluatex.dtx v1.0n	
General: Undefine XeTeX classes	General: contribute_filter added .	489
when using patching an older	insert_local_par added	489
kernel	2016/03/29 ltpictur.dtx v.1l	
2016/01/05 ltfinal.dtx v2.0p	\@oval: add setting of line tests .	335
General: Only apply XeTeX	initialise tests	334
change if XeTeX is in use . .	\@ovhorz: use glue not leaders if	
2016/02/11 ltluatex.dtx v1.0m	horizontal line not required .	335
General: pdf_stream_filter_callback	\@ovvert: use glue not leaders if	
removed	vertical line not required . . .	335
process_rule, [hv]pack_quality	\if@ovhline: macro added	
append_to_vlist_filter added .	(latex/4452)	334
read_cidmap_file added	\if@ovvline: macro added	
show_warning_message added .	(latex/4452)	334

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

$\!$ b361, b363, z144, O257
 $\"$ 1176, 1305, 1343, 1381, 1390,
 1463, 1495, 1522, 1530, 1536, 1540,
 1546, 1550, 1556, 1562, 1569, 1570,
 1576, 1580, 1630, 1673, o350, O258
 $\#$ a57, a70,
 b6, b14, b428, d314, o337, O241
 $\$$ a69, b4, b13, d313, l256,
 l368, l375, l452, l685, l692, O242
 $\%$ a70, a100,
 a102, a122, b14, b426, d314,
 l413, l415, o339, L499, L500, O243
 $\&$ a69, b5, b13, b427, d313, L118, O244
 $\'$. b448, 1177, 1306, 1345, 1379, 1387,
 1465, 1475, 1481, 1483, 1486, 1488,
 1496, 1502, 1508, 1510, 1513, 1515,
 1523, 1527, 1534, 1538, 1543, 1548,
 1551, 1553, 1560, 1565, 1566, 1573,
 1578, 1581, 1631, 1675, 1694, 1696,
 1697, 1698, 1701, 1703, 1704, 1705,
 1707, 1708, o349, s168, t172,
 y145, z151, B236, C61, K558, O259
 \langle z168, z238
 \rangle b448, z168, z239
 \ast o342, z148, L433, L501
 $\+$ C61
 $\,$. b362, b364, i281, t414, y145, z7,
 z8, z40, z108, z110, z113, z127, z144
 $\-$ b330, d9, d11,
 i272, l340, l341, l458, l669, l670,
 o344, y145, B235, C61, O151, O197
 $\.$ b361, b363, k39, l178, l307,
 l376, l377, l396, l471, l472, l498,
 l499, l525, l632, l699, l706, o343
 \surd a92, d12, o291, o345, L117
 $\:$ b362, b364, d306, d307, z149
 $\;$ b362, b364, t408, z128, z144
 \lt l459, l623, o340, y145, C60, C98
 \simeq 1179, l308, l395, l633, s168, B236, C60
 \gt 1456, l624, o341, y145, z144, z149, C60
 $\?$ b361, b363, O259
 $\@$ a60, d308, d309, g19,
 i284, j2, L24, L32, N18, N732, O251
 $\@@$ a302, a303, f15, f19, f20,
 f21, f22, f24, f27, f28, f30, f31,
 k203, k219, p464, p466, p467,
 C199, C200, C201, C211, K10, K11
 $\@defaultsubs$ o440
 $\@enc@update$ 1133, o224, o228
 $\@end$.. a64, a217, d8, k183, k184,
 s135, y39, y49, M18, O347, O368
 $\@endpbox$ C166, C197, C345
 $\@eqnocr$ z273, z291, z294, z399
 $\@fileswith@ptions$ L191, L361
 $\@hyph$ d9
 $\@hyphenation$ l155
 $\@if@newlist$ K568, K615
 $\@if@definable$ d109, l17
 $\@input$ a63, d7, k162, k163, k172, y19
 $\@italiccorr$ d12, v96, v100
 $\@line$ B368
 $\@math@bgroup$ v114, v121
 $\@math@egroup$ v111, v111
 $\@par$ 64, d6, h4, y49,
 y104, y108, y111, A82, A85,
 B217, B234, C172, F50, F101, K231
 $\@patterns$ l155
 $\@protect$ d231, d237, d246
 $\@startpbox$ C166, C197, C345
 $\@underline$ B325, B328, B329
 $\@unprocessedoptions$... L343, L405
 $\@warning$ g170
 $\@Alph$ m49, m65
 $\@DeclareMathDelimiter$... r678, r697
 $\@DeclareMathSizes$.. o171, o172, o174
 $\@Esphack$ i110, G201, G223, G241
 $\@IncludeInRelease$ c57, c58
 $\@IncludeInRelease$ c55, c56, c57
 $\@M$ b21, b405, b406, d24,
 d26, i27, i28, i29, i30, i31, i32,
 i33, i34, i57, p393, p406, z283,
 A194, C56, F50, F83, F101,
 F113, F154, K171, K172, K232
 $\@MM$ b21, G418, K273
 $\@Mi$ e3, K143
 $\@Mii$.. e3, G53, G122, G194, G216,
 G241, G315, K269, K1069, K1236
 $\@Miii$ e3, G55, G124, G317, K272
 $\@Miv$ e6, G195, G201, G217, G223, K246
 $\@Roman$ m47, m53
 $\@TeXversion$ 1, 6, a297, g28
 $\@acci$ s168, B236

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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx

- \@accii [s168](#), [B236](#)
- \@acciii [s168](#), [B236](#)
- \@acol [C141](#),
[C151](#), [C221](#), [C222](#), [C234](#), [C235](#),
[C238](#), [C255](#), [C268](#), [C276](#), [C286](#)
- \@acolampacol [C219](#), [C236](#),
[C238](#), [C245](#), [C253](#), [C285](#), [C288](#)
- \@activechar@info [K549](#)
- \@addamp [C212](#), [C221](#),
[C222](#), [C237](#), [C251](#), [C286](#), [C287](#)
- \@addfield [C43](#),
[C53](#), [C75](#), [C82](#), [C114](#), [C125](#), [C127](#)
- \@addmarginpar [K305](#), [K1721](#)
- \@addtobot [K885](#), [K972](#),
[K1039](#), [K1091](#), [K1200](#), [K1259](#)
- \@addtocurcol [K302](#), [K976](#), [K1875](#)
- \@addtodblcol [K764](#), [K1472](#)
- \@addtofilelist [a96](#), [a98](#), [k54](#), [k162](#),
[k200](#), [s125](#), [s143](#), [s147](#), [s154](#),
[s157](#), [s164](#), [s167](#), [O210](#), [O213](#), [O388](#)
- \@addtonextcol .. [K763](#), [K1296](#), [K1876](#)
- \@addtopreamble [C270](#), [C283](#),
[C289](#), [C290](#), [C291](#), [C293](#), [C305](#)
- \@addtoreset [m16](#), [m39](#), [m44](#)
- \@addtotoporbot [K922](#),
[K1085](#), [K1253](#), [K1345](#), [K1434](#)
- \@afterheading [F75](#), [F108](#)
- \@afterindentfalse [F28](#)
- \@afterindenttrue .. [F26](#), [F107](#), [F153](#)
- \@alph [m48](#), [m61](#), [G379](#)
- \@ampacol ... [C219](#), [C236](#), [C247](#), [C288](#)
- \@arabic [m43](#), [m45](#), [m51](#), [G377](#)
- \@argarraycr [C176](#), [C177](#)
- \@argdef [d57](#)
- \@argrbox [B348](#)
- \@argtabularcr [C183](#), [C184](#)
- \@array [C154](#), [C155](#)
- \@arrayacol [C141](#), [C219](#)
- \@arrayclassiv [C142](#), [C290](#)
- \@arrayclassz [C141](#), [C236](#)
- \@arraycr [C143](#), [C174](#), [C176](#)
- \@arrayparboxrestore [B231](#), [B245](#), [C343](#)
- \@arrayrule [C268](#),
[C270](#), [C274](#), [C276](#), [C278](#), [C305](#)
- \@arstrut [C165](#), [C198](#), [C302](#)
- \@arstrutbox . [C158](#), [C191](#), [C302](#), [C344](#)
- \@author [F5](#)
- \@auxout [k81](#), [k87](#), [k103](#), [k118](#),
[x33](#), [F145](#), [I7](#), [I8](#), [I19](#), [I29](#), [I37](#), [I43](#)
- \@backslashchar
..... [d195](#), [g189](#), [g191](#), [t185](#), [L466](#)
- \@badcrrr [g231](#)
- \@badend [g202](#), [y65](#)
- File Key: [a=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=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- \@badlinearg [g221](#), [D58](#),
[D67](#), [D68](#), [D72](#), [D116](#), [D121](#), [D132](#)
- \@badmath .. [g205](#), [z172](#), [z174](#), [z179](#),
[z181](#), [z189](#), [z201](#), [z206](#), [z215](#),
[z227](#), [z232](#), [z323](#), [z335](#), [z351](#), [z360](#)
- \@badpoptabs [g209](#), [C74](#), [C136](#)
- \@badrequireerror [L131](#), [L413](#)
- \@badtab [g212](#),
[C22](#), [C76](#), [C97](#), [C103](#), [C110](#), [C133](#)
- \@begin@tempboxa
[B26](#), [B41](#), [B155](#), [B217](#), [B349](#), [B357](#)
- \@begin@documenthook
..... [k48](#), [k51](#), [L372](#), [L386](#), [I33](#)
- \@begin@divi [K594](#), [K620](#)
- \@begin@divibox [K93](#), [K621](#)
- \@begin@parpenalty
..... [i30](#), [z326](#), [z338](#), [z364](#), [A23](#), [A170](#)
- \@begin@theorem [E30](#), [E35](#)
- \@bezier [D368](#), [D369](#)
- \@bibitem [I3](#), [I8](#)
- \@biblabel [I4](#), [I54](#)
- \@bitor [K15](#),
[K791](#), [K811](#), [K847](#), [K870](#), [K937](#),
[K1021](#), [K1031](#), [K1179](#), [K1190](#),
[K1332](#), [K1419](#), [K1537](#), [K1662](#)
- \@botlist [K65](#),
[K358](#), [K360](#), [K405](#), [K407](#), [K627](#),
[K648](#), [K657](#), [K658](#), [K899](#), [K902](#),
[K937](#), [K939](#), [K1031](#), [K1033](#),
[K1190](#), [K1192](#), [K1832](#), [K1859](#)
- \@botnum [G278](#),
[K116](#), [K896](#), [K897](#), [K902](#), [K906](#),
[K913](#), [K1368](#), [K1373](#), [K1461](#),
[K1468](#), [K1824](#), [K1851](#), [K1893](#)
- \@botroom [G279](#),
[K117](#), [K899](#), [K902](#), [K1825](#), [K1852](#)
- \@boxfpsbit [K1941](#), [K1943](#), [K1948](#)
- \@break@tfor [f31](#), [k157](#), [v81](#)
- \@bsphack [i9](#), [i63](#), [i232](#),
[i248](#), [x32](#), [G52](#), [G121](#), [G314](#),
[H6](#), [H18](#), [H23](#), [H35](#), [K1792](#), [I39](#)
- \@caption [G12](#), [G14](#)
- \@capttype [G5](#), [G9](#),
[G12](#), [G40](#), [G88](#), [G109](#), [G157](#), [K1905](#)
- \@car [36](#), [d40](#), [j14](#), [I77](#)
- \@carcube [d42](#), [d112](#)
- \@ccclv [b16](#), [K274](#), [K278](#),
[K356](#), [K357](#), [K386](#), [K403](#), [K404](#),
[K433](#), [K457](#), [K461](#), [K462](#), [O28](#), [O47](#)
- \@ccclvi [b21](#), [b57](#), [b77](#), [b86](#),
[b88](#), [b92](#), [b152](#), [b166](#), [N30](#), [N58](#)
- \@cdr [36](#), [d40](#), [d287](#), [d288](#)
- \@centercr [y68](#), [y76](#), [y83](#), [y89](#)

- \@centering z253,
z254, z261, z264, z267, z392, z396
 - \@cflb K624
 - \@cflt K624
 - \@changed@cmd .. l3, l63, l173, o96, o232
 - \@changed@x l3, l161, l169
 - \@changed@x@mouth l161, l169
 - \@charlb k121, k129
 - \@charrb k123, k129
 - \@chclass C232, C233, C294, C307, C312
 - \@check@c d166, d168
 - \@check@eq d172, d173, d177
 - \@checkend y11, y61, y64
 - \@chnum C240,
C259, C294, C309, C310, C311
 - \@circ D340, D341, D342
 - \@circle D328, D329
 - \@circlefnt D37, D39, D232,
D261, D303, D333, D348, D363
 - \@cite l16, l52
 - \@cite@ofmt l24, l53
 - \@citea l15, l17
 - \@citeb l16, l18, l19,
l20, l23, l24, l41, l42, l43, l44, l45
 - \@citex l13, l14
 - \@classi C232, C266
 - \@classii C232, C280
 - \@classiii C232, C285
 - \@classiv C142, C153, C233
 - \@classoptionslist
l9, l160, l171, l288, l289, l524
 - \@classv C233, C291
 - \@classz C141, C152, C232
 - \@cline C326
 - \@clnht D74, D75, D83,
D85, D87, D97, D104, D130, D357
 - \@clnwd D76, D82, D86, D88, D89, D357
 - \@cls@pkg L94, L95,
L323, L352, L389, L398, L400, L417
 - \@clsextension
... l16, l41, l52, l70, l101,
l127, l144, l160, l170, l210,
l225, l233, l287, l356, l364, l390
 - \@clubpenalty
. k9, k19, A128, A196, F89, F118
 - \@colht k16, G277,
G279, G282, G288, G289,
G302, G303, K121, K205, K216,
K225, K226, K361, K373, K408,
K421, K448, K479, K509, K515,
K519, K529, K534, K616, K687,
K725, K769, K794, K813, K853,
K875, K1552, K1678, K2006, O82
 - \@colnum G280, K118,
K905, K950, K1019, K1020,
K1048, K1056, K1098, K1177,
K1178, K1210, K1222, K1266,
K1330, K1331, K1368, K1373,
K1417, K1418, K1460, K1467,
K1820, K1847, K1886, K2061
 - \@colroom k17,
K122, K226, K247, K248, K259,
K262, K361, K408, K687, K904,
K949, K1015, K1018, K1047,
K1172, K1176, K1209, K1326,
K1329, K1412, K1416, K1821,
K1848, K2016, K2021, K2066, O81
 - \@combinedblfloats K660, K2119, K2158
 - \@combinefloats K475, K624
 - \@comdblflleft K660
 - \@comfleleft K630, K646, K660
 - \@cons 36, b189, b206, d39,
m44, G193, G215, G239, G359,
K211, K798, K817, K833, K857,
K859, K879, K881, K1051,
K1119, K1215, K1288, K1361,
K1451, K1554, K1577, K1680,
K1705, K1722, K1723, K2067
 - \@contfield C50, C126, C138
 - \@ctrerr .. g198, m64, m68, m82, m90
 - \@curfield C16, C41,
C47, C51, C52, C54, C119, C120
 - \@curline
C16, C27, C39, C44, C53, C54,
C55, C79, C80, C92, C117, C118
 - \@curr@enc l114, l116
 - \@currbox b256, b257, b258,
G60, G91, G95, G129, G160,
G164, G193, G214, G215,
G239, G261, G263, G265,
G323, G326, G331, G335, K187,
K188, K199, K200, K202, K203,
K211, K285, K286, K763, K764,
K1012, K1014, K1022, K1045,
K1049, K1051, K1066, K1107,
K1119, K1167, K1170, K1207,
K1212, K1215, K1232, K1277,
K1288, K1320, K1336, K1350,
K1361, K1403, K1440, K1451,
K1491, K1495, K1506, K1512,
K1514, K1518, K1523, K1532,
K1541, K1547, K1554, K1577,
K1612, K1616, K1628, K1635,
K1637, K1641, K1647, K1657,
K1672, K1680, K1705, K1723,
K1732, K1911, K1912, K1941,
K1971, K1976, K2022, K2025,
- File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- [K2037](#), [K2045](#), [K2062](#), [K2067](#)
`\@currdir` [1](#), [6](#), [a103](#), [a125](#), [a127](#), [a133](#),
[a135](#), [a141](#), [a143](#), [a148](#), [a150](#),
[a160](#), [a173](#), [a238](#), [a251](#), [a264](#), [L443](#)
`\@current@cmd` [l25](#), [o236](#)
`\@currentlabel` [x34](#),
[x37](#), [x40](#), [z257](#), [z377](#), [B298](#), [G420](#)
`\@currenvir`
[g203](#), [y3](#), [y55](#), [y65](#), [A112](#), [B102](#),
[L460](#), [L466](#), [L474](#), [L478](#), [L484](#)
`\@currentline` ... [g203](#), [y56](#), [y66](#), [B103](#)
`\@currentext` [L15](#), [L23](#), [L31](#), [L100](#), [L101](#),
[L144](#), [L153](#), [L160](#), [L170](#), [L220](#),
[L229](#), [L314](#), [L315](#), [L320](#), [L321](#),
[L326](#), [L332](#), [L336](#), [L338](#), [L340](#),
[L342](#), [L344](#), [L345](#), [L348](#), [L354](#),
[L356](#), [L364](#), [L382](#), [L390](#), [L406](#), [L407](#)
`\@currlist`
[G193](#), [G215](#), [G359](#), [K67](#), [K285](#),
[K362](#), [K365](#), [K409](#), [K412](#), [K1722](#)
`\@currname` [c60](#), [c67](#),
[k211](#), [k212](#), [L14](#), [L22](#), [L30](#), [L92](#),
[L94](#), [L100](#), [L153](#), [L229](#), [L313](#),
[L315](#), [L338](#), [L340](#), [L342](#), [L344](#),
[L345](#), [L382](#), [L398](#), [L400](#), [L407](#), [L417](#)
`\@currnamestack` [L20](#)
`\@curroptions`
... [L153](#), [L161](#), [L183](#), [L407](#), [L408](#)
`\@currsize` [s72](#)
`\@currtype` [K126](#),
[K788](#), [K789](#), [K790](#), [K791](#), [K808](#),
[K809](#), [K810](#), [K811](#), [K937](#),
[K1021](#), [K1031](#), [K1179](#), [K1190](#),
[K1332](#), [K1419](#), [K1537](#), [K1662](#),
[K1911](#), [K1913](#), [K1914](#), [K1917](#)
`\@currtab` [C11](#),
[C26](#), [C75](#), [C76](#), [C77](#), [C83](#), [C84](#),
[C87](#), [C91](#), [C92](#), [C96](#), [C131](#), [C132](#)
`\@currtabmar` [C11](#), [C25](#),
[C26](#), [C38](#), [C44](#), [C78](#), [C91](#), [C95](#), [C96](#)
`\@d@r` [a156](#), [a157](#)
`\@dashbox` [D175](#), [D176](#),
[D177](#), [D178](#), [D179](#), [D182](#), [D185](#),
[D187](#), [D196](#), [D198](#), [D199](#), [D200](#),
[D201](#), [D204](#), [D207](#), [D210](#), [D359](#)
`\@dashcnt` [D169](#), [D170](#),
[D171](#), [D172](#), [D173](#), [D174](#), [D184](#),
[D186](#), [D189](#), [D190](#), [D191](#), [D192](#),
[D194](#), [D195](#), [D206](#), [D209](#), [D359](#)
`\@dashdim` [D168](#), [D169](#), [D170](#),
[D171](#), [D173](#), [D176](#), [D178](#), [D179](#),
[D180](#), [D184](#), [D186](#), [D188](#), [D189](#),
[D190](#), [D191](#), [D194](#), [D198](#), [D200](#),
[D201](#), [D202](#), [D208](#), [D211](#), [D359](#)
`\@date` [F7](#)
`\@dbflt` [G32](#), [G268](#)
`\@dblarg` [35](#), [d311](#), [F37](#), [F125](#), [G12](#)
`\@dbldeferlist` [G239](#),
[K74](#), [K419](#), [K424](#), [K426](#), [K726](#),
[K733](#), [K734](#), [K1662](#), [K1665](#),
[K1705](#), [K1707](#), [K1836](#), [K1864](#)
`\@dblfloat` [G31](#)
`\@dblfloatplacement`
[k25](#), [G275](#), [G284](#), [K375](#), [K423](#),
[K1817](#), [K1844](#), [K2124](#), [K2164](#)
`\@dblflset` [G26](#)
`\@dblfpbot` [G294](#), [G308](#), [K2208](#)
`\@dblfpsep` [G293](#), [G307](#), [K2208](#)
`\@dblfpstop` [G292](#), [G306](#), [K2208](#)
`\@dbltoplist`
[K69](#), [K206](#), [K209](#), [K211](#), [K371](#),
[K372](#), [K419](#), [K420](#), [K665](#), [K669](#),
[K671](#), [K672](#), [K1549](#), [K1554](#),
[K1674](#), [K1680](#), [K1835](#), [K1862](#)
`\@dbltopnum` [G287](#), [G301](#),
[K114](#), [K134](#), [K212](#), [K214](#), [K676](#),
[K1488](#), [K1489](#), [K1553](#), [K1556](#),
[K1564](#), [K1584](#), [K1589](#), [K1609](#),
[K1610](#), [K1679](#), [K1683](#), [K1691](#),
[K1712](#), [K1717](#), [K1828](#), [K1855](#)
`\@dbltoproom` .. [G288](#), [G290](#), [G302](#),
[G304](#), [K115](#), [K1491](#), [K1494](#),
[K1495](#), [K1504](#), [K1505](#), [K1508](#),
[K1511](#), [K1514](#), [K1518](#), [K1522](#),
[K1526](#), [K1531](#), [K1551](#), [K1612](#),
[K1615](#), [K1616](#), [K1625](#), [K1626](#),
[K1627](#), [K1630](#), [K1634](#), [K1637](#),
[K1641](#), [K1646](#), [K1650](#), [K1655](#),
[K1656](#), [K1677](#), [K1829](#), [K1856](#)
`\@dec@text@cmd` [l3](#)
`\@declaredoptions`
[L8](#), [L134](#), [L157](#), [L173](#), [L188](#), [L370](#)
`\@declareoption` ... [L132](#), [L133](#), [L141](#)
`\@defaultsubs` .. [o394](#), [o428](#), [o440](#), [y26](#)
`\@defaultunits` [o179](#), [o183](#),
[o184](#), [o185](#), [o200](#), [o262](#), [p133](#), [p135](#)
`\@defdefault@ds` ... [L132](#), [L137](#), [L142](#)
`\@deferlist` [K68](#),
[K74](#), [K358](#), [K367](#), [K368](#), [K371](#),
[K376](#), [K378](#), [K384](#), [K405](#), [K414](#),
[K416](#), [K688](#), [K696](#), [K697](#), [K708](#),
[K713](#), [K714](#), [K1021](#), [K1024](#),
[K1119](#), [K1121](#), [K1179](#), [K1182](#),
[K1288](#), [K1290](#), [K1332](#), [K1334](#),
[K1361](#), [K1363](#), [K1419](#), [K1421](#),
[K1451](#), [K1453](#), [K1537](#), [K1539](#),
[K1577](#), [K1579](#), [K1834](#), [K1861](#)
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- `\definecounter`
 . [m12](#), [m36](#), [z242](#), [A227](#), [A228](#),
 [A229](#), [A230](#), [E8](#), [E16](#), [G376](#), [G378](#)
- `\depth` [d13](#), [p145](#),
 [t464](#), [t465](#), [t467](#), [t468](#), [B324](#),
 [B367](#), [C160](#), [C192](#), [D106](#), [D157](#),
 [D160](#), [D175](#), [D182](#), [D402](#), [K1761](#)
- `\dir` [a155](#), [a158](#), [a160](#), [a162](#), [a163](#)
- `\dischyph` [d11](#), [B235](#)
- `\docclearpage` [K270](#), [K345](#)
- `\documentclasshook` [L3](#), [L292](#)
- `\doendpe` [y62](#), [A123](#)
- `\dofilelist` [k209](#), [k225](#), [y21](#)
- `\donoparitem` [A144](#), [A158](#)
- `\dot` [D328](#), [D341](#)
- `\dotsep` [F160](#)
- `\dottedtocline` [F149](#)
- `\downline` [D154](#), [D158](#), [D163](#)
- `\downvector` [D125](#), [D163](#)
- `\eha` [d255](#), [g174](#), [g192](#),
 [g194](#), [g196](#), [g204](#), [g206](#), [g236](#),
 [k88](#), [l52](#), [l979](#), [l989](#), [o25](#), [o67](#),
 [o109](#), [o152](#), [o218](#), [o273](#), [p106](#),
 [r25](#), [r70](#), [r99](#), [r161](#), [r192](#), [r293](#),
 [r314](#), [r346](#), [r387](#), [r432](#), [r437](#),
 [r492](#), [r601](#), [r605](#), [r609](#), [r644](#),
 [r654](#), [r739](#), [r744](#), [r747](#), [r779](#),
 [r782](#), [r837](#), [r840](#), [r843](#), [r910](#),
 [r916](#), [v129](#), [y54](#), [K1786](#), [K1802](#), [I47](#)
- `\ehb` [g174](#), [g199](#), [g224](#),
 [g226](#), [g228](#), [K208](#), [K364](#), [K411](#)
- `\ehc` [d105](#),
 [d132](#), [g174](#), [g231](#), [g234](#), [g240](#),
 [g242](#), [y130](#), [y141](#), [z298](#), [A220](#), [F4](#)
- `\ehd` . [g174](#), [g201](#), [g208](#), [g211](#), [g213](#),
 [g219](#), [r118](#), [C89](#), [C98](#), [G6](#), [L258](#)
- `\elt` [d39](#), [k122](#), [m20](#), [m35](#), [K8](#),
 [K11](#), [K15](#), [K27](#), [K30](#), [K31](#), [K32](#),
 [K33](#), [K38](#), [K39](#), [K40](#), [K41](#), [K42](#),
 [K43](#), [K44](#), [K45](#), [K47](#), [K51](#), [K57](#),
 [K58](#), [K59](#), [K60](#), [K472](#), [K630](#),
 [K641](#), [K646](#), [K656](#), [K668](#), [K670](#),
 [K698](#), [K715](#), [K735](#), [K754](#), [K767](#),
 [K774](#), [K825](#), [K828](#), [K837](#), [K1808](#)
- `\empty` [f14](#)
- `\emptycol`
 . . . [K172](#), [K219](#), [K222](#), [K251](#), [K255](#)
- `\end@tempboxa`
 . . . [B35](#), [B44](#), [B160](#), [B230](#), [B355](#), [B365](#)
- `\enddocumenthook` . . . [y10](#), [L372](#), [L387](#)
- `\endfloatbox` [G190](#), [G211](#), [G236](#), [G248](#)
- `\endparenv` [A120](#), [A123](#)
- `\endparpenalty`
 . . . [i31](#), [z327](#), [z339](#), [z365](#), [A23](#), [A124](#)
- File Key:** [a=ltmdirchk.dtx](#), [b=ltplain.dtx](#), [c=ltvers.dtx](#), [d=ltdefs.dtx](#),
[e=ltalloc.dtx](#), [f=ltcntrl.dtx](#), [g=lterror.dtx](#), [h=ltpar.dtx](#), [i=ltspace.dtx](#),
[j=ltlogos.dtx](#), [k=ltfiles.dtx](#), [l=ltoutenc.dtx](#), [m=ltcounts.dtx](#), [n=ltlength.dtx](#),
[o=ltfssbas.dtx](#), [p=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- `\endpbox` [C166](#),
 [C197](#), [C227](#), [C292](#), [C343](#), [C346](#)
- `\endpefalse` [y59](#), [A129](#),
 [A131](#), [A135](#), [A136](#), [A138](#), [B105](#)
- `\endpeltrue` [A138](#)
- `\endpetrue` [A124](#), [A126](#), [A134](#)
- `\endtheorem` [E13](#), [E19](#), [E25](#), [E35](#)
- `\enlargepage` . . . [K1771](#), [K1776](#), [K1778](#)
- `\ensuredmath` [z309](#), [z311](#)
- `\enumctr` [A234](#), [A237](#), [A238](#)
- `\enumdepth` . . . [A226](#), [A232](#), [A233](#), [A234](#)
- `\eqcnt` [z250](#),
 [z295](#), [z300](#), [z379](#), [z394](#), [z395](#), [z397](#)
- `\eqncr` . . . [z262](#), [z280](#), [z301](#), [z302](#), [z381](#)
- `\eqnnum` . . . [z244](#), [z245](#), [z299](#), [z313](#), [z372](#)
- `\eqnset` [z250](#), [z393](#)
- `\eqnswfalse` [z279](#)
- `\eqnswtrue` [z252](#), [z258](#), [z300](#), [z378](#)
- `\eqpen` [z250](#), [z283](#), [z285](#), [z292](#)
- `\err@` [g37](#),
 [g41](#), [g44](#), [g52](#), [g64](#), [g68](#), [g71](#), [g79](#)
- `\esphack` . [i11](#), [i69](#), [i237](#), [i254](#), [x35](#),
 [G365](#), [H17](#), [H19](#), [H34](#), [K1794](#), [I50](#)
- `\evenfoot` [J12](#), [J15](#), [K584](#)
- `\evenhead` [J12](#), [J15](#), [K583](#)
- `\expandtwoargs`
 [d193](#), [L74](#), [L159](#), [L173](#), [L197](#)
- `\expast` [C200](#), [C228](#)
- `\failedlist`
 . . . [K752](#), [K775](#), [K791](#), [K798](#),
 [K811](#), [K817](#), [K833](#), [K847](#), [K870](#)
- `\fcolmadefalse` [K743](#)
- `\fcolmadetrue` [K831](#)
- `\filef@und` . . . [k144](#), [k154](#), [k162](#), [k172](#)
- `\filelist`
 . . . [k53](#), [k199](#), [k200](#), [k211](#), [s125](#),
 [s143](#), [s154](#), [s164](#), [O210](#), [O372](#), [O388](#)
- `\fileswfalse` [k64](#)
- `\fileswith@ptions` [L131](#),
 [L191](#), [L283](#), [L284](#), [L286](#), [L310](#), [L361](#)
- `\fileswith@ptions`
 [L278](#), [L279](#), [L281](#), [L285](#)
- `\fileswithoptions`
 [L210](#), [L217](#), [L225](#), [L276](#)
- `\fileswtrue` [k7](#)
- `\finalstrut` . [B302](#), [B366](#), [C344](#), [G425](#)
- `\firstampfalse` . . . [C215](#), [C238](#), [C255](#)
- `\firstamptrue` [C223](#)
- `\firstcolfirstmark`
 [K2101](#), [K2102](#), [K2106](#)
- `\firstcoltopmark` [K2099](#), [K2107](#)
- `\firstcolumnfalse` [K2091](#), [K2136](#)
- `\firstcolumntrue`
 [k22](#), [K105](#), [K181](#), [K2110](#), [K2142](#)

- \@firstofone [d188](#), [k47](#),
[l68](#), [l113](#), [p300](#), [r53](#), [r81](#), [r142](#),
[r172](#), [r689](#), [y9](#), [z307](#), [C331](#), [G10](#),
[N67](#), [N103](#), [N111](#), [N169](#), [I18](#), [I42](#)
 - \@firstoftwo [a82](#), [d188](#), [d283](#), [d310](#),
[k155](#), [l97](#), [l951](#), [l967](#), [m100](#),
[m105](#), [r693](#), [x19](#), [J16](#), [L48](#), [L64](#), [L77](#)
 - \@firstttab [C2](#), [C63](#), [C64](#), [C65](#), [C95](#), [C107](#)
 - \@flcheckspace ... [K899](#), [K935](#), [K2012](#)
 - \@flfail [K775](#),
[K826](#), [K847](#), [K857](#), [K870](#), [K879](#)
 - \@float [G26](#), [G32](#)
 - \@floatboxreset ... [G101](#), [G170](#), [G174](#)
 - \@floatpenalty
... [G3](#), [G53](#), [G55](#), [G58](#), [G122](#),
[G124](#), [G127](#), [G191](#), [G194](#),
[G199](#), [G201](#), [G212](#), [G216](#),
[G221](#), [G223](#), [G237](#), [G241](#),
[G315](#), [G317](#), [G321](#), [G325](#), [G359](#)
 - \@floatplacement [k25](#), [G275](#), [K156](#),
[K183](#), [K227](#), [K451](#), [K1818](#), [K1845](#)
 - \@flsetnum [K896](#),
[K932](#), [K1019](#), [K1177](#), [K1330](#),
[K1417](#), [K1488](#), [K1609](#), [K1980](#)
 - \@flsettextmin [K995](#),
[K1147](#), [K1316](#), [K1399](#), [K1996](#)
 - \@flstop [K1882](#)
 - \@flsucceed
.. [K768](#), [K776](#), [K825](#), [K859](#), [K881](#)
 - \@fltovf [g227](#), [G93](#), [G162](#), [G326](#)
 - \@flupdates [K902](#), [K947](#), [K2058](#)
 - \@flushglue
[e17](#), [y77](#), [y83](#), [y90](#), [y103](#), [A76](#), [B242](#)
 - \@fnsymbol [m50](#), [m69](#)
 - \@font@info [o98](#), [o136](#), [o142](#),
[o300](#), [o317](#), [o476](#), [p30](#), [p38](#), [p46](#),
[p74](#), [p87](#), [p126](#), [p154](#), [p168](#),
[p179](#), [p193](#), [p209](#), [p215](#), [p228](#),
[p235](#), [p242](#), [p247](#), [p257](#), [p269](#),
[p281](#), [p445](#), [p457](#), [p462](#), [p469](#),
[p494](#), [p502](#), [r202](#), [r217](#), [r251](#),
[r297](#), [r366](#), [r372](#), [r416](#), [r429](#),
[r512](#), [r592](#), [r635](#), [r729](#), [r878](#), [r907](#)
 - \@font@warning [o4](#), [o390](#), [o395](#), [o422](#),
[o429](#), [p19](#), [p33](#), [p41](#), [p49](#), [p61](#),
[p77](#), [p430](#), [p444](#), [p456](#), [p461](#),
[p468](#), [p493](#), [p501](#), [q30](#), [y23](#), [O216](#)
 - \@fontswitch [v109](#), [v111](#)
 - \@footnotemark
.. [G407](#), [G413](#), [G431](#), [G437](#), [G438](#)
 - \@footnotetext [B272](#),
[G407](#), [G413](#), [G414](#), [G447](#), [G453](#)
 - \@for [f16](#), [k99](#), [k211](#), [L78](#), [L157](#), [L171](#),
[L183](#), [L188](#), [L203](#), [L408](#), [I16](#), [I41](#)
 - \@forloop [f19](#), [f20](#)
 - \@fornoop [f15](#), [f23](#), [f29](#)
 - \@fortmp [f17](#), [f18](#), [f26](#)
 - \@fpbot [G294](#), [G308](#), [K773](#), [K2202](#)
 - \@fpmin [G282](#), [G291](#), [G305](#),
[K120](#), [K830](#), [K1826](#), [K1853](#), [K2073](#)
 - \@fps [G41](#), [G42](#), [G44](#),
[G47](#), [G64](#), [G110](#), [G111](#), [G113](#),
[G116](#), [G133](#), [K1903](#), [K1905](#), [K1908](#)
 - \@fpsaddddefault
.. [G45](#), [G48](#), [G114](#), [G117](#), [K1900](#)
 - \@fpsep [G293](#), [G307](#),
[K771](#), [K780](#), [K852](#), [K874](#), [K2202](#)
 - \@fpstype [K893](#), [K914](#), [K915](#), [K929](#),
[K960](#), [K961](#), [K985](#), [K987](#), [K990](#),
[K992](#), [K1043](#), [K1099](#), [K1100](#),
[K1135](#), [K1138](#), [K1141](#), [K1144](#),
[K1205](#), [K1267](#), [K1268](#), [K1306](#),
[K1308](#), [K1311](#), [K1313](#), [K1387](#),
[K1390](#), [K1393](#), [K1396](#), [K1485](#),
[K1500](#), [K1502](#), [K1520](#), [K1529](#),
[K1565](#), [K1566](#), [K1606](#), [K1621](#),
[K1623](#), [K1643](#), [K1653](#), [K1692](#),
[K1693](#), [K1896](#), [K1912](#), [K1914](#),
[K1916](#), [K1919](#), [K1920](#), [K1921](#),
[K1923](#), [K1924](#), [K1928](#), [K1929](#),
[K1931](#), [K1932](#), [K1966](#), [K1968](#),
[K1970](#), [K1982](#), [K1984](#), [K1998](#),
[K2000](#), [K2030](#), [K2033](#), [K2044](#)
 - \@fptop [G292](#), [G306](#), [K770](#), [K2202](#)
 - \@frameb@x [B132](#), [B159](#), [B161](#)
 - \@framebox [B139](#), [B146](#), [B149](#)
 - \@framepicbox [B139](#), [B146](#), [B182](#)
 - \@freelist . [b189](#), [b206](#), [b256](#), [G60](#),
[G129](#), [G323](#), [G324](#), [K29](#), [K34](#),
[K48](#), [K56](#), [K187](#), [K473](#), [K642](#),
[K657](#), [K671](#), [K776](#), [K1722](#), [K1723](#)
 - \@getcirc ... [D222](#), [D255](#), [D299](#), [D331](#)
 - \@getfpsbit
[K890](#), [K926](#), [K1482](#), [K1603](#), [K1939](#)
 - \@getlarrow [D123](#), [D131](#), [D133](#)
 - \@getlinechar [D69](#), [D108](#)
 - \@getpen [i7](#), [i10](#), [i21](#), [i55](#)
 - \@getrarrow [D124](#), [D131](#), [D140](#)
 - \@glossaryfile [H21](#), [H22](#), [H31](#)
 - \@gnewline [i46](#), [i48](#), [i49](#)
 - \@gobble ... [d88](#), [d110](#), [d185](#), [d195](#),
[d213](#), [d217](#), [d252](#), [d258](#), [d261](#),
[d270](#), [f6](#), [f9](#), [g101](#), [g127](#), [g153](#),
[g162](#), [i42](#), [i312](#), [k54](#), [k199](#), [l29](#),
[l929](#), [o391](#), [o424](#), [p299](#), [q26](#), [r28](#),
[r30](#), [r255](#), [r266](#), [r330](#), [r377](#), [r378](#),
[r407](#), [r413](#), [r421](#), [r426](#), [r444](#),
[r458](#), [r468](#), [r477](#), [r490](#), [r507](#),
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- r516, r586, r595, r629, r638,
- r722, r732, r795, r800, r869,
- r900, s147, s157, s167, F126,
- F127, F128, F129, F130, F146,
- G7, K590, K591, K592, K837,
- K1810, K2074, L246, L429,
- L453, L458, N66, N101, O213,
- O317, O323, O388, I11, I25, I26
- \@gobble@IncludeInRelease c64, c71, c74
- \@gobblecr i310, i311
- \@gobblefour d185,
- r24, r252, r368, r370, r374,
- r376, r386, r390, r514, r566, L460
- \@gobbletwo d152, d153,
- d185, f12, k26, o396, o430, r132,
- y16, y24, J11, J13, L452, O222
- \@gtempa
d103, d104, d158, d160, k180,
k181, k183, k184, k185, C3, C5,
C6, C7, C8, L91, L92, L102, L104
- \@halfwidth D2, D38,
D40, D41, D106, D156, D159,
D175, D182, D196, D206, D209,
D365, D387, D400, D401, D402
- \@halignto .. C143, C147, C150, C164
- \@hangfrom F49, F100, F121
- \@height b399, d13, i242,
i250, l242, l244, p144, t246,
t464, t465, t467, t468, B116,
B121, B168, B178, B324, B367,
C159, C192, C318, C335, D106,
D157, D160, D175, D182, D198,
D205, D280, D323, D401, K1761
- \@highpenalty i56, O3
- \@hightab ... C11, C21, C23, C63,
C75, C84, C85, C100, C131, C132
- \@hline D60, D105, D122
- \@holdpg K129, K274,
K276, K277, K282, K283, K284
- \@hspace i296, i297
- \@hspace i296, i298
- \@hvector D118, D122
- \@icentercr y71, y72
- \@iden d191
- \@if d148, d149, d151
- \@if@ptions L74, L76, L89
- \@if@ptions .. L69, L70, L73, L75, L321
- \@ifatmargin C55, C95
- \@ifclasslater 455, L51
- \@ifclassloaded 455, L40
- \@ifclasswith 455, L69
- \@ifdefinable 35, d61,
d63, d107, d109, d215, l14, l17,
m11, n3, s68, B69, E7, E15, E22
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- \@iffileonpath k140, k148
- \@ifl@aded .. L40, L41, L44, L50, L320
- \@ifl@t@r L56, L59, L66, L267
- \@ifl@ter 1912,
1913, L51, L52, L55, L58, L348
- \@ifl@ter@@ 1912, 1913
- \@ifnch d293, d295, d307
- \@ifnextchar
... 35, a93, d289, d294, d310,
i44, i311, k163, m13, p365,
y70, z248, A143, B9, B11, B18,
B20, B25, B46, B75, B76, B82,
B83, B89, B93, B138, B139,
B145, B146, B150, B183, B191,
B199, B205, B209, B250, B254,
B258, B309, B314, B336, B343,
B347, C57, C154, C176, C183,
D10, D42, D53, D238, E3, E5,
E28, G27, G268, G328, G405,
G428, G445, K183, K1884,
L97, L262, L277, L282, I3, I13
- \@iforloop f21, f22
- \@ifpackagelater 455, L51
- \@ifpackageloaded .. 455, K1868, L40
- \@ifpackagewith 455, L69
- \@ifframebox B151, B152, B153
- \@ifframepicbox B183, B184
- \@ifstar 35, d50, d310, i38, i226, i296,
o171, q121, y69, y136, z282,
C56, C175, C182, D52, D328,
F35, F125, K1766, L132, L154
- \@ifundefined 35, d104,
d111, d131, d138, d160, d171,
d252, d258, d281, l931, m3, m7,
m16, o65, o151, p378, r287, x23,
y44, y53, E21, J3, J7, L38,
L122, L184, L490, L493, I20, I44
- \@ignorefalse y4, y58, y63, G364
- \@ignoretrue i120,
i133, y4, y7, z241, z244, z276, z402
- \@iiminipage
.. B252, B256, B259, B260, B261
- \@iiparbox B193, B201,
B207, B210, B211, B212, B289
- \@iiminipage B255, B257
- \@iinput k163, k164
- \@iiparbox B206, B208
- \@iirsbox B347, B356
- \@imakebox B25, B40, B91
- \@imakepicbox ... B46, B47, B96, B185
- \@iminipage B251, B253
- \@include k89, k90
- \@index H18, H19, H35
- \@indexfile H4, H5, H14

- `\@inlabelfalse` [A28](#), [A104](#), [A184](#), [K166](#)
`\@inlabeltrue` [A28](#), [A178](#)
`\@inmatherr` [g237](#), [A112](#), [A142](#), [D328](#)
`\@inmathwarn` [l3](#)
`\@input` [k28](#), [k93](#), [k171](#), [F135](#)
`\@input@` [k108](#), [k173](#), [o327](#), [I31](#)
`\@inputcheck` [a65](#), [a186](#), [a187](#), [a190](#), [a198](#),
[d25](#), [d32](#), [k3](#), [k135](#), [k136](#), [k143](#),
[k152](#), [k153](#), [k156](#), [L440](#), [L441](#), [L448](#)
`\@insertfalse` [K983](#), [K1133](#),
[K1304](#), [K1385](#), [K1480](#), [K1601](#)
`\@inserttrue` [K909](#), [K954](#),
[K1071](#), [K1239](#), [K1559](#), [K1686](#)
`\@invalidchar` [g242](#)
`\@iparbox` [B192](#), [B200](#), [B204](#)
`\@irsbox` [B336](#), [B343](#), [B347](#), [B348](#)
`\@isavebox` [B89](#), [B90](#)
`\@isavepicbox` [B94](#), [B95](#)
`\@ishortstack` [D43](#), [D51](#)
`\@istackcr` [D53](#), [D54](#)
`\@itabcr` [C57](#), [C58](#)
`\@item` [A143](#), [A156](#)
`\@itemdepth` [A241](#), [A243](#), [A244](#), [A245](#)
`\@itemfudge` [C38](#), [C44](#), [C71](#)
`\@itemitem` [A245](#), [A248](#)
`\@itemlabel` [A44](#), [A96](#), [A143](#)
`\@itempenalty` [i32](#), [A23](#), [A175](#)
`\@iwhiledim` [f7](#)
`\@iwhilenum` [f3](#)
`\@iwhilesw` [f10](#)
`\@ixpt` [o504](#)
`\@ixstackcr` [D52](#)
`\@killglue` [D22](#), [D30](#), [D36](#)
`\@kludgeins` [K293](#),
[K294](#), [K295](#), [K297](#), [K350](#), [K351](#),
[K397](#), [K398](#), [K476](#), [K492](#), [K493](#),
[K499](#), [K500](#), [K501](#), [K510](#),
[K526](#), [K530](#), [K540](#), [K1762](#), [K1793](#)
`\@labels` [A146](#), [A147](#), [A189](#), [A206](#), [A207](#)
`\@largefloatcheck` [G192](#), [G213](#), [G238](#), [G260](#)
`\@lastchclass` [C223](#),
[C233](#), [C234](#), [C236](#), [C244](#), [C267](#),
[C281](#), [C285](#), [C294](#), [C307](#), [C308](#)
`\@latex@error` [d105](#),
[d132](#), [d253](#), [g136](#), [g172](#), [g188](#),
[g194](#), [g196](#), [g199](#), [g201](#), [g203](#),
[g206](#), [g208](#), [g210](#), [g213](#), [g217](#),
[g222](#), [g226](#), [g228](#), [g230](#), [g231](#),
[g233](#), [g236](#), [g240](#), [g242](#), [k88](#), [l50](#),
[o6](#), [o25](#), [o67](#), [o109](#), [o152](#), [o218](#),
[o273](#), [p105](#), [q100](#), [q111](#), [r23](#),
[r68](#), [r97](#), [r117](#), [r159](#), [r190](#), [r213](#),
[r229](#), [r293](#), [r314](#), [r346](#), [r386](#),
[r390](#), [r432](#), [r437](#), [r492](#), [r560](#),
[r566](#), [r601](#), [r605](#), [r609](#), [r644](#),
[r654](#), [r739](#), [r744](#), [r747](#), [r779](#),
[r782](#), [r837](#), [r840](#), [r843](#), [r910](#),
[r916](#), [s50](#), [s100](#), [v126](#), [y54](#), [y129](#),
[y141](#), [z298](#), [A219](#), [C89](#), [C98](#),
[F4](#), [G6](#), [G83](#), [L221](#), [L240](#), [L253](#),
[L322](#), [L397](#), [L414](#), [L422](#), [L427](#), [I47](#)
`\@latex@info` [d201](#), [d272](#), [g136](#)
`\@latex@info@no@line` [g136](#), [K550](#)
`\@latex@warning` [g136](#), [g170](#), [l55](#), [x14](#), [D234](#),
[G264](#), [K1906](#), [L477](#), [L483](#), [I22](#), [I45](#)
`\@latex@warning@no@line` [d179](#), [g136](#), [g171](#),
[k13](#), [k197](#), [x8](#), [x26](#), [x27](#), [y31](#),
[F6](#), [K217](#), [K249](#), [K1737](#), [K1972](#),
[L93](#), [L268](#), [L349](#), [L442](#), [L449](#), [L507](#)
`\@latexbug` [g229](#), [K307](#), [K1723](#)
`\@latexerr` [g170](#),
[K208](#), [K364](#), [K411](#), [K1784](#), [K1801](#)
`\@lbibitem` [I3](#), [I4](#)
`\@ldots` [t412](#), [t414](#)
`\@leftcolumn` [K128](#),
[K2092](#), [K2113](#), [K2137](#), [K2146](#)
`\@leftmark` [J16](#), [J36](#)
`\@let@token` [d293](#),
[d296](#), [d299](#), [d307](#), [i266](#), [i267](#),
[i274](#), [v66](#), [v79](#), [z153](#), [z155](#), [z158](#)
`\@lign` [z138](#), [z140](#)
`\@linechar` [D69](#),
[D70](#), [D71](#), [D75](#), [D76](#), [D78](#), [D83](#),
[D85](#), [D86](#), [D87](#), [D88](#), [D90](#), [D94](#),
[D95](#), [D98](#), [D99](#), [D104](#), [D129](#), [D355](#)
`\@linefnt` [D37](#), [D39](#), [D69](#),
[D122](#), [D130](#), [D161](#), [D164](#), [D362](#)
`\@linelen` [D57](#),
[D58](#), [D82](#), [D89](#), [D98](#), [D100](#),
[D105](#), [D106](#), [D107](#), [D115](#), [D116](#),
[D157](#), [D160](#), [D162](#), [D163](#), [D356](#)
`\@listctr` [A202](#), [A225](#), [I9](#)
`\@listdepth` [A23](#), [A35](#), [A38](#), [A43](#), [A99](#), [B273](#)
`\@listfiles` [k52](#), [k203](#), [k218](#)
`\@loadwithoptions` [L227](#), [L233](#), [L237](#)
`\@lowpenalty` [i55](#), [O3](#)
`\@ltab` [C60](#), [C95](#)
`\@m` [b21](#), [b359](#),
[b361](#), [b362](#), [b395](#), [b396](#), [i184](#),
[i288](#), [i293](#), [k39](#), [A80](#), [D92](#), [D96](#), [I17](#)
`\@mainaux` [k5](#), [k31](#), [k32](#), [k81](#), [k93](#), [k118](#), [y15](#)
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- \@makebox B11, B20, [B24](#)
 - \@makecaption G24
 - \@makecol [K235](#), [K387](#), [K434](#), [K454](#)
 - \@makefcolumn
 - [K367](#), [K368](#), [K376](#), [K378](#),
 - [K414](#), [K416](#), [K424](#), [K426](#), [K2071](#)
 - \@makefnmark [G380](#), [G441](#)
 - \@makefntext [B301](#), [G424](#)
 - \@makefurther a71,
 - a92, a121, d313, d314, o340,
 - o341, o342, o343, o344, o345,
 - o346, o347, o348, o349, o350,
 - y113, [y123](#), [y134](#), [L117](#), [L118](#), [L465](#)
 - \@makepicbox B10, B19, [B45](#), [D211](#)
 - \@makespecialcolbox [K477](#), [K496](#)
 - \@marbox [G324](#), [G326](#), [G330](#), [G334](#),
 - [G335](#), [G359](#), [K1722](#), [K1732](#),
 - [K1735](#), [K1743](#), [K1745](#), [K1746](#),
 - [K1748](#), [K1749](#), [K1750](#), [K1759](#)
 - \@marginparreset [G343](#), [G350](#)
 - \@markright [J29](#), [J34](#)
 - \@maxdepth [k50](#), [K98](#), [K460](#), [K488](#), [O79](#)
 - \@maxtab [C2](#), [C83](#)
 - \@medpenalty [i56](#), [O3](#)
 - \@midlist
 - [K66](#), [K473](#), [K474](#), [K937](#), [K939](#),
 - [K1051](#), [K1215](#), [K1833](#), [K1860](#)
 - \@minipagefalse [A181](#), [B246](#),
 - [B248](#), [B286](#), [G187](#), [G250](#), [G345](#)
 - \@minipagerestore [B274](#), [B276](#)
 - \@minipagetrue [B247](#), [G186](#)
 - \@minus [d13](#), [K2195](#),
 - [K2196](#), [K2197](#), [K2200](#), [K2201](#)
 - \@missingfileerror
 - [456](#), [k167](#), [k174](#), [L342](#)
 - \@miv [e3](#)
 - \@mkboth [J11](#), [J13](#)
 - \@mklab [A45](#), [A140](#)
 - \@mkpream [C162](#), [C195](#), [C223](#)
 - \@mparbottom [G367](#),
 - [G368](#), [K125](#), [K450](#), [K1733](#),
 - [K1741](#), [K1742](#), [K1743](#), [K1744](#)
 - \@mpargs [B265](#), [B289](#)
 - \@mparswitchfalse [K109](#)
 - \@mpfn [B271](#), [G405](#), [G410](#), [G450](#), [G454](#)
 - \@mpfootins [B280](#),
 - [B281](#), [B284](#), [B290](#), [B293](#), [B294](#)
 - \@mpfootnotetext [B272](#), [B292](#)
 - \@mplistdepth [B273](#), [B290](#)
 - \@multicnt
 - [C329](#), [C331](#), [C332](#), [C333](#), [C340](#),
 - [C341](#), [C342](#), [D30](#), [D31](#), [D33](#),
 - [D352](#), [D385](#), [D387](#), [D388](#), [D389](#),
 - [D390](#), [D394](#), [D398](#), [D409](#), [D413](#)
 - \@multiplelabels [k27](#), [x25](#), [x31](#), [y29](#), [y35](#)
 - \@multiput [D28](#), [D29](#)
 - \@multispan [C330](#), [C334](#), [C338](#)
 - \@namedef [35](#), [d37](#), [k128](#),
 - [1934](#), [o100](#), [o101](#), [o125](#), [p372](#),
 - [x28](#), [y121](#), [z302](#), [z303](#), [C148](#),
 - [E12](#), [E13](#), [E18](#), [E19](#), [E23](#), [E24](#), [E25](#)
 - \@nameuse
 - [35](#), [d38](#), [k116](#), [k127](#), [E23](#), [J5](#), [K578](#)
 - \@nbitem [A168](#), [A221](#)
 - \@ne [b16](#)
 - \@needsf@rmat [L263](#), [L266](#), [L271](#)
 - \@needsformat [L251](#), [L261](#), [L265](#)
 - \@negargfalse [D65](#)
 - \@negargtrue [D64](#)
 - \@newcommand [d56](#), [d57](#)
 - \@newctr [m13](#), [m15](#), [E8](#)
 - \@newenv [d127](#), [d128](#), [d137](#)
 - \@newenva [d125](#), [d126](#)
 - \@newenvb [d127](#), [d128](#)
 - \@newl@bel [x22](#), [y17](#), [I10](#)
 - \@newline [i45](#), [i47](#)
 - \@newlistfalse
 - [A29](#), [A33](#), [A108](#), [A182](#), [K569](#)
 - \@newlisttrue [A29](#), [A33](#), [A87](#)
 - \@next [b256](#),
 - [G60](#), [G129](#), [G323](#), [G324](#), [K9](#),
 - [K187](#), [K285](#), [K787](#), [K807](#), [K1722](#)
 - \@nextchar
 - [C230](#), [C231](#), [C289](#), [C290](#), [C291](#)
 - \@nil [a156](#), [a157](#), [c12](#),
 - [c18](#), [c61](#), [c62](#), [d40](#), [d41](#), [d42](#),
 - [d112](#), [d287](#), [d288](#), [f13](#), [f19](#), [f27](#),
 - [j14](#), [l77](#), [o292](#), [o303](#), [o356](#), [o459](#),
 - [o462](#), [o463](#), [o471](#), [p304](#), [p305](#),
 - [p307](#), [p320](#), [p326](#), [p330](#), [p331](#),
 - [p367](#), [p388](#), [p393](#), [p473](#), [p487](#),
 - [q26](#), [q44](#), [q53](#), [q57](#), [r40](#), [r356](#),
 - [r364](#), [r397](#), [r921](#), [r923](#), [v41](#), [v45](#),
 - [C326](#), [C327](#), [L27](#), [L29](#), [L60](#),
 - [L61](#), [L67](#), [L202](#), [L205](#), [L299](#), [L307](#)
 - \@nmbriistfalse [A33](#), [A46](#), [A91](#)
 - \@nmbriisttrue [A225](#)
 - \@nnil [f13](#), [f20](#), [f21](#),
 - [f22](#), [f28](#), [o179](#), [o183](#), [o184](#), [o185](#),
 - [o200](#), [p133](#), [p135](#), [p299](#), [p301](#),
 - [p313](#), [p315](#), [p320](#), [p334](#), [p336](#),
 - [p343](#), [p354](#), [p355](#), [p357](#), [p388](#), [p393](#)
 - \@no@font@optfalse [q17](#), [q129](#)
 - \@no@lnbk [i13](#), [i14](#), [i15](#)
 - \@no@pgbk [i3](#), [i4](#), [i5](#)
 - \@nobreakfalse
 - [i58](#), [i60](#), [A193](#), [F77](#), [F112](#),
 - [F140](#), [G182](#), [K168](#), [K1060](#), [K1226](#)
- File Key: a=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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx

- `\@nobreaktrue` i59, F109, G181
`\@nocnterr` g195
`\@nocounterr` g195, m4, m8, m16, E21
`\@nodocument` g200,
k58, y50, G39, G108, K159, K186
`\@noitemargfalse` A32, A200
`\@noitemargtrue` A32, A143
`\@noitemerr` g232,
i164, i199, i222, A69, A81, A107
`\@noligs` y114, y135, y151
`\@nolnerr` g193, i17, i51, y68
`\@nomath` o2, o271, s35, s42, s63, s65, s70
`\@noparitemfalse` A30, A145
`\@noparitemtrue` A30, A66
`\@noparlistfalse` A31, A70
`\@noparlisttrue` A31, A67
`\@normalcr` i35, i43, B245
`\@normalsize` L4, L5
`\@noskipsecfalse` k45, F81, K161
`\@noskipsectrue` F21, F78
`\@notdefinable` d113, d114, d118, g187
`\@notprerr` g235, k56
`\@nthm` E3, E4
`\@nxttabmar` C11, C21, C23,
C25, C64, C100, C101, C107, C108
`\@obsoletefile` k196
`\@oddfoot` J11, J14, J15, K131, K581
`\@oddhead` J11, J14, K130, K581
`\@onefilewithoptions`
. . . L291, L295, L301, L311, L360
`\@onelevel@sanitize` d315, G42, G111
`\@onlypreamble` d43, d165, d167,
d176, d184, k61, k70, k85, k198,
k224, l23, l24, l61, l62, l66,
l89, l109, l139, l140, l154, l935,
o18, o80, o82, o88, o104, o132,
o147, o168, o173, o215, o367,
p373, q28, q36, q42, q79, q83,
q88, q93, q98, q108, q126, q127,
q128, q134, q138, q142, r17, r19,
r44, r46, r107, r116, r136, r243,
r244, r247, r279, r317, r319,
r321, r334, r349, r396, r398,
r440, r479, r495, r572, r612,
r615, r657, r660, r663, r683,
r696, r750, r785, r789, r792,
r846, r866, r870, r934, v123,
v124, x30, H12, H29, L10, L12,
L18, L19, L26, L28, L34, L36,
L39, L42, L43, L50, L53, L54,
L58, L66, L68, L71, L72, L75,
L89, L98, L106, L108, L125,
L128, L129, L140, L141, L142,
L149, L155, L168, L181, L193,
L195, L200, L206, L211, L215,
L218, L226, L231, L234, L238,
L247, L260, L265, L271, L280,
L285, L310, L360, L362, L371,
L384, L385, L388, L395, L404,
L411, L412, L420, L425, L430,
L513, L514, L515, L516, L518, I40
`\@opargbegintheorem` E32, E35
`\@opcol` K236, K244,
K368, K387, K416, K434, K439
`\@options` L194
`\@othm` E3, E20
`\@outerparskip`
. A8, A88, A117, A152, A222
`\@outputbox` K127, K457, K459,
K479, K482, K483, K503, K505,
K506, K511, K514, K519, K521,
K528, K534, K536, K607, K633,
K639, K649, K650, K673, K680,
K766, K769, K772, K778, K779,
K2092, K2096, K2097, K2111,
K2117, K2137, K2143, K2152
`\@outputdblcol` K442,
K2087, K2089, K2133, K2134
`\@outputpage`
. K377, K426, K444, K564,
K2121, K2126, K2159, K2167
`\@oval` D238, D246, D291
`\@ovbtrue` D247, D292
`\@ovdx` D216,
D257, D259, D265, D267, D279,
D282, D301, D309, D311, D322,
D324, D374, D375, D376,
D377, D391, D392, D394, D408
`\@ovdy` D216,
D258, D260, D266, D267, D272,
D277, D302, D310, D311, D316,
D320, D381, D382, D383,
D384, D395, D396, D398, D412
`\@ovhlinefalse` D248
`\@ovhlinetrue`
. D241, D245, D253, D259, D287
`\@ovhorz` D264,
D265, D278, D308, D309, D321
`\@ovltrue` D247, D292
`\@ovri` B32, D216, D256,
D272, D283, D300, D316, D325
`\@ovro` D216,
D256, D265, D266, D271, D277,
D278, D300, D309, D310,
D315, D320, D321, D332, D339
`\@ovrtrue` D247, D292
`\@ovttrue` D247, D292
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- \@ovvert D262,
D263, [D269](#), D304, D306, D313
- \@ovvlinefalse D248
- \@ovvlinetrue D244, D252, D260
- \@ovvxx [D216](#), D250, D252,
D253, D257, D263, D264, D278,
D295, D297, D301, D306, D308,
D321, D371, D372, D373, D377,
D386, D387, D393, D394, D407
- \@ovvy [D216](#),
D251, D252, D253, D258, D265,
D269, D296, D297, D302, D309,
D313, D378, D379, D380,
D384, D386, D397, D398, D411
- \@p@pfilename L27, L29, L34
- \@pagedp K124, K282, K287,
K1001, K1154, K1751, K1761
- \@pageht K123, K283,
K287, K289, K290, K291, K295,
K1000, K1153, K1734, K1741
- \@par [h3](#), [h6](#)
- \@parboxrestore B217,
B245, B270, B297, G19, G100,
G169, G342, G419, K193, K570
- \@parboxto [B212](#)
- \@parmoderr ... [g225](#), G58, G127, G320
- \@parse@version
.... [c61](#), [c62](#), L60, L61, L67, L68
- \@partaux [k5](#), [k87](#), [k103](#),
[k105](#), [k106](#), [k112](#), [k121](#), [k123](#), [k126](#)
- \@partlist [k84](#), [k99](#)
- \@partswfalse [k8](#)
- \@partswtrue [k83](#)
- \@pass@ptions
... [L120](#), [L125](#), [L126](#), [L127](#), [L336](#)
- \@pboxswfalse B215, B263
- \@pboxswtrue B225
- \@penup [z129](#), [z130](#)
- \@percentchar [a101](#),
[L457](#), [L459](#), [L461](#), [L463](#), [L502](#)
- \@picbox [D6](#), D13, D19, D20
- \@picht [D6](#), D12, D19
- \@picture D10, [D11](#)
- \@picture@warn [D102](#), [D226](#), [D230](#), [D234](#)
- \@pkgextension [L16](#), L40, L51, L69,
[L126](#), [L217](#), [L220](#), [L237](#), [L302](#), [L406](#)
- \@plus [d13](#), [i302](#), F16, F151,
J40, K2195, K2196, K2197,
K2200, K2201, K2205, K2206,
K2207, K2211, K2212, K2213
- \@pnumwidth F163
- \@popfilename [L20](#), [L357](#)
- \@pr@videpackage [L97](#), [L99](#), [L106](#)
- File Key: a=lt`dirchk`.dtx, b=lt`plain`.dtx, c=lt`vers`.dtx, d=lt`defs`.dtx,
e=lt`alloc`.dtx, f=lt`cntrl`.dtx, g=lt`error`.dtx, h=lt`par`.dtx, i=lt`space`.dtx,
j=lt`logos`.dtx, k=lt`files`.dtx, l=lt`outenc`.dtx, m=lt`counts`.dtx, n=lt`length`.dtx,
o=lt`fssbas`.dtx, p=lt`fssstrc`.dtx, q=lt`fssc`.dtx, r=lt`fssdcl`.dtx, s=lt`fssini`.dtx,
t=lt`fontdef`.dtx, u=lt`preload`.dtx, v=lt`fontcmd`.dtx, w=lt`pageno`.dtx, x=lt`xref`.dtx,
y=lt`miscen`.dtx, z=lt`math`.dtx, A=lt`lists`.dtx, B=lt`boxes`.dtx, C=lt`tab`.dtx,
D=lt`pictur`.dtx, E=lt`thm`.dtx, F=lt`ssect`.dtx, G=lt`float`.dtx, H=lt`idxglo`.dtx,
I=lt`bibl`.dtx, J=lt`page`.dtx, K=lt`output`.dtx, L=lt`class`.dtx, M=lt`thyphen`.dtx,
N=lt`luatex`.dtx, O=lt`final`.dtx
- \@preamble C163, C165,
C173, C198, C217, C219, C220,
C224, C239, C257, C258, C293
- \@preamblecmds .. [d43](#), [k57](#), L525, L526
- \@preamerr ... [g214](#), C172, C235, C314
- \@process@pti@ns
..... L167, L180, L182, L193
- \@process@ptions .. L154, L156, L168
- \@protected@testopt [d66](#), [d78](#)
- \@providesfile .. [a93](#), [a94](#), [L109](#), [O384](#)
- \@ptionlist
..... [L37](#), [L74](#), [L153](#), [L326](#), [L332](#), [L407](#)
- \@pushfilename [L20](#), [L312](#)
- \@put [D237](#), D267, D311, D339
- \@qend [d113](#), [d287](#), [g191](#)
- \@qrelax [d114](#), [d287](#)
- \@rc@ifdefinable [d107](#), [d109](#), [d215](#), [l14](#)
- \@reargdef [d99](#)
- \@refundefined [k46](#), [x3](#), [y27](#)
- \@reinserts K301, K304, [K490](#)
- \@removeelement [f32](#), [L197](#)
- \@reqcolroom K1000, K1001,
K1004, K1006, K1007, K1012,
K1016, K1018, K1046, K1047,
K1153, K1154, K1158, K1161,
K1162, K1167, K1174, K1176,
K1208, K1209, K1320, K1322,
K1324, K1327, K1329, K1403,
K1406, K1409, K1414, K1416,
[K1896](#), K2013, K2018, K2021
- \@reset@ptions L317, L358, [L363](#)
- \@resetactivechars [K549](#), [K567](#)
- \@resethfps K1115, K1284, [K1963](#)
- \@restorepar
... [64](#), [h6](#), [i233](#), [i249](#), A127, A135
- \@reversemarginfalse ... G368, K108
- \@reversemargintrue G367
- \@rightmark [J16](#), [J37](#)
- \@rightskip [y79](#), [y83](#), A75, B241
- \@rjfieldfalse C34, C66
- \@rjfieldtrue C114
- \@roman [m46](#), [m52](#)
- \@rsbox B336, B343, [B346](#)
- \@rtab C60, [C75](#)
- \@rule B309, B314, [B317](#)
- \@sanitize .. [d313](#), H7, H18, H24, H35
- \@savebox B76, B83, [B88](#)
- \@savemarbox . G330, G331, G334, [G337](#)
- \@savepicbox B76, B83, [B92](#)
- \@savsf [i61](#), [i67](#), [i76](#), [i91](#), [i103](#), [i117](#), [i131](#)
- \@savsk [i61](#), [i66](#), [i77](#), [i92](#), [i104](#), [i118](#), [i132](#)
- \@scolelt K698, [K763](#)
- \@sdblcolelt K715, [K735](#), [K764](#)
- \@secntformat F43, [F94](#)

- \@secondoftwo [a83](#), [d188](#),
[d285](#), [k149](#), [l95](#), [l953](#), [l969](#), [m99](#),
[m104](#), [x21](#), [J17](#), [L46](#), [L62](#), [L84](#)
- \@secpenalty [i33](#), [F19](#), [F33](#)
- \@sect [F37](#), [F38](#)
- \@seqnrcr [z301](#)
- \@setckpt [k121](#), [k128](#), [y16](#)
- \@setfloattyperecounts
..... [K984](#), [K1134](#), [K1305](#),
[K1386](#), [K1481](#), [K1602](#), [K1910](#)
- \@setfontsize [s70](#)
- \@setfps [G34](#)
- \@setfpsbit [G73](#), [G75](#), [G77](#),
[G85](#), [G143](#), [G146](#), [G149](#), [K1954](#)
- \@setmarks [K2103](#), [K2105](#), [K2120](#)
- \@setminipage
.. [B275](#), [G21](#), [G177](#), [G185](#), [G356](#)
- \@setnobreak [G179](#), [G355](#)
- \@setpar [64](#), [h3](#), [A78](#)
- \@setref [x10](#)
- \@setsize [s70](#)
- \@settab [C60](#), [C82](#)
- \@settodim [n17](#)
- \@settopoint [n22](#)
- \@sharp .. [C169](#), [C196](#), [C226](#), [C241](#),
[C242](#), [C260](#), [C262](#), [C264](#), [C292](#)
- \@shipoutsetup [K564](#)
- \@shortstack [D42](#), [D43](#)
- \@sline [D60](#), [D63](#), [D126](#)
- \@slowromancap [m53](#), [m54](#)
- \@spaces [g173](#)
- \@specialoutput [K230](#)
- \@specialpagefalse [K104](#), [K578](#)
- \@specialpagetrue [J9](#)
- \@specialstyle [J9](#), [K578](#)
- \@sptoken [d296](#), [d306](#)
- \@sqrt [z248](#)
- \@ssect [F36](#), [F95](#)
- \@stackcr [D49](#), [D52](#)
- \@star@or@long [d49](#), [d54](#),
[d101](#), [d123](#), [d129](#), [d155](#), [d164](#), [d198](#)
- \@startcolumn [K237](#), [K244](#), [K685](#)
- \@startdblcolumn [K685](#),
[K2125](#), [K2128](#), [K2165](#), [K2171](#)
- \@startfield
. [C28](#), [C46](#), [C81](#), [C93](#), [C114](#), [C122](#)
- \@startline .. [C20](#), [C57](#), [C58](#), [C59](#), [C72](#)
- \@startpbox [C166](#),
[C197](#), [C227](#), [C291](#), [C343](#), [C345](#)
- \@startsection [F22](#)
- \@starttoc [F132](#)
- \@stopfield [C32](#), [C48](#), [C59](#),
[C75](#), [C82](#), [C114](#), [C116](#), [C125](#), [C127](#)
- \@stopline [C30](#), [C56](#), [C74](#)
- File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- \@stpelt [m20](#), [m23](#)
- \@strip@args [l74](#)
- \@svector [D118](#), [D126](#)
- \@sverb [y136](#), [y137](#), [y144](#)
- \@svsec [F40](#), [F43](#), [F49](#), [F61](#)
- \@svsechd [F59](#), [F84](#), [F104](#)
- \@sxverbatim [y95](#), [y121](#)
- \@tabacckludge .. [l173](#), [l175](#), [l378](#), [l379](#)
- \@tabacol [C151](#), [C219](#)
- \@tabarray [C143](#), [C153](#), [C154](#)
- \@tabclassiv [C153](#), [C289](#)
- \@tabclassz [C152](#), [C243](#)
- \@tabcr [C56](#), [C62](#)
- \@tabfbox [C16](#), [C69](#), [C71](#)
- \@tablab [C61](#), [C115](#)
- \@tabminus [C61](#), [C106](#)
- \@tabplus [C61](#), [C99](#)
- \@tabpush
[C11](#), [C66](#), [C74](#), [C125](#), [C128](#), [C130](#)
- \@tabrj [C61](#), [C113](#)
- \@tabular [C147](#), [C150](#), [C151](#)
- \@tabularcr [C153](#), [C181](#)
- \@tempboxa
[e13](#), [l69](#), [n17](#), [n18](#), [A205](#), [A211](#),
[A212](#), [A214](#), [B28](#), [B29](#), [B30](#),
[B31](#), [B36](#), [B37](#), [B38](#), [B39](#), [B128](#),
[B157](#), [B164](#), [B174](#), [B266](#), [B289](#),
[B352](#), [B353](#), [B354](#), [B361](#), [B362](#),
[B363](#), [B364](#), [D161](#), [D162](#), [D232](#),
[D233](#), [D256](#), [D261](#), [D266](#), [D267](#),
[D300](#), [D303](#), [D310](#), [D311](#), [D332](#),
[D333](#), [D338](#), [D339](#), [D399](#), [D417](#),
[F121](#), [F122](#), [G326](#), [G360](#), [K279](#),
[K351](#), [K356](#), [K357](#), [K398](#), [K403](#),
[K404](#), [K540](#), [K597](#), [K604](#), [K605](#),
[K631](#), [K635](#), [K647](#), [K653](#), [K660](#),
[K661](#), [K662](#), [K663](#), [K667](#), [K675](#)
- \@tempcnta ... [e7](#), [r665](#), [r666](#), [r667](#),
[r668](#), [r672](#), [C203](#), [C204](#), [C205](#),
[C206](#), [D66](#), [D67](#), [D93](#), [D94](#),
[D95](#), [D108](#), [D109](#), [D110](#), [D111](#),
[D113](#), [D114](#), [D127](#), [D128](#), [D133](#),
[D135](#), [D136](#), [D137](#), [D138](#), [D139](#),
[D142](#), [D144](#), [D145](#), [D146](#), [D147](#),
[D148](#), [D149](#), [D150](#), [D151](#), [D152](#),
[D153](#), [D183](#), [D184](#), [D185](#), [D186](#),
[D187](#), [D205](#), [D206](#), [D207](#), [D208](#),
[D209](#), [D210](#), [D223](#), [D224](#), [D225](#),
[D227](#), [D229](#), [D231](#), [D233](#), [D270](#),
[D275](#), [D314](#), [D318](#), [D334](#), [D335](#),
[D336](#), [D337](#), [D343](#), [D344](#), [D345](#),
[D346](#), [D347](#), [D348](#), [D390](#), [D406](#),
[G62](#), [G68](#), [G70](#), [G79](#), [G80](#),
[G90](#), [G91](#), [G131](#), [G137](#), [G139](#),

- G152, G153, G159, G160,
K16, K18, K20, K844, K845,
K846, K847, K867, K868, K869,
K870, K892, K895, K928, K931,
K1042, K1204, K1484, K1487,
K1605, K1608, K1723, K1725,
K1728, K1730, K1732, K1754,
K1944, K1945, K1949, K1955,
K1959, O154, O159, O160,
O161, O229, O234, O235, O236
- `\@tempcntb` . . . [e7](#), r666, r670, r672,
D136, D137, D138, D140, D141,
D142, D270, D271, D275, D276,
D314, D315, D318, D319, G88,
G89, G90, G157, G158, G159,
K17, K20, K21, K1955, K1956,
K1957, O155, O159, O230, O234
- `\@tempdima` . [e10](#), o184, o189, z116,
z119, z125, B42, B43, B156,
B157, B162, B163, B164, B166,
B216, B217, B264, B268, B320,
B323, B324, B350, B352, B358,
B361, C35, C36, C37, C77,
C78, C79, C80, C191, C192,
D89, D90, D92, D93, D94,
D95, D96, D97, D222, D223,
D224, D233, D257, D258, D262,
D263, D301, D302, D304, D306,
D335, D337, D342, D343, D344,
F156, F157, F166, G196, G198,
G218, G220, G262, G263,
G264, K203, K204, K205, K461,
K463, K509, K511, K512, K517,
K522, K526, K531, K535, K827,
K830, K850, K860, K872, K882,
K1547, K1548, K1551, K1552,
K1672, K1673, K1677, K1678,
K1733, K1734, K1735, K1736,
K1739, K1742, K1745, K1747,
K2062, K2063, K2065, K2066
- `\@tempdimb` [e10](#), o185,
o190, o479, o483, p133, p134,
p391, p414, p415, p424, p425,
p429, p447, p450, p453, p455,
B219, B220, B321, B324, B351,
B353, B359, B362, D90, D91,
D252, D254, D255, D297, D298,
D299, D330, D331, D340, D341,
K850, K851, K852, K853, K860,
K872, K873, K874, K875, K882
- `\@tempdimc` . [e10](#), p408, p409, p411,
p412, p414, p415, B322, B323, B324
- `\@tempskipa` . [e14](#), i19, i22, i23, i181,
i188, i190, i193, p135, p136,
- A116, A117, A118, A150, A152,
A153, A154, A222, A223, A224,
F25, F27, F28, F33, F45, F46,
F71, F72, F74, F86, F87, F96,
F97, K1782, K1783, K1785, K1793
- `\@tempskipb` . . . [e14](#), i140, i142, i144,
i147, i149, i159, i179, i181, i182,
i186, i188, i190, i191, i214, i217
- `\@tempswafalse` a73, b244,
k97, o59, r281, r336, r400, r481,
r909, r915, y18, y105, K898,
K934, K1490, K1611, L438, I13
- `\@tempswatru` a74, b249, k95, k100, o62, r284,
r339, r403, r484, r872, y42,
y110, K1492, K1515, K1613,
K1638, K2023, K2040, L437, I13
- `\@temptokena` [e16](#), y45,
y46, J22, J23, J30, J31, J34, J35
- `\@testdef` [y17](#), [y40](#)
- `\@testfalse` K12, K14, K15
- `\@testfp` K792,
K812, K848, K871, [K1947](#), K2074
- `\@testopt` d20, d56,
[d76](#), d80, d125, i3, i4, i13, i14, z288
- `\@testpach` C231, C307
- `\@testpatch` [C307](#)
- `\@testtrue` K13, K21, K330,
K795, K814, K854, K876, K1951
- `\@testwrongwidth` [K319](#),
K793, K849, K1022, K1336, K1541
- `\@text@composite` [174](#)
- `\@text@composite@x` [174](#)
- `\@textbottom` J40, J42, K485, K523, K537, [K546](#)
- `\@textfloatsheight`
. K450, K997, K999, K1049,
K1050, K1055, K1150, K1152,
K1212, K1214, K1220, [K1896](#)
- `\@textmin` G289, G290, G303, G304,
K119, K999, K1003, K1006,
K1007, K1152, K1157, K1161,
K1162, K1324, K1409, K1508,
K1510, K1526, K1630, K1632,
K1650, K2004, K2006, K2008
- `\@textsubscript`
. G391, [G392](#), G399, G402
- `\@textsuperscript` . G381, G383, [G384](#)
- `\@texttop` . J40, J42, K481, K504, [K546](#)
- `\@tf@r` f25, f26
- `\@tf@r` [f25](#), k150, k205, v71,
B51, C229, D249, D293, G63, K132
- `\@tf@r@loop` f27, f28, f30
- `\@thanks` F10, [F13](#)
- File Key:** a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- \@thefnmark B299,
G380, G381, G406, G411,
G421, G430, G435, G446, G451
 - \@thefoot K131, K581, K584, K611
 - \@thehead K130, K581, K583, K601
 - \@themargin K81, K582, K584, K596
 - \@themark J21, J22, J29, J30, J35, J38
 - \@thirdofthree d192, l147
 - \@thm E12, E18, E24, E26
 - \@thmcounter E11, E17, E33
 - \@thmcountersep E10, E33
 - \@title F3
 - \@tocrmarg F152
 - \@toodeep g207, A36, A232, A243
 - \@toplist K64, K358, K359,
K405, K406, K626, K632, K642,
K643, K935, K947, K1831, K1858
 - \@topnewpage K173
 - \@topnum G275,
K112, K932, K933, K947, K951,
K959, K1368, K1373, K1461,
K1468, K1822, K1849, K1890
 - \@toproom G277,
K113, K935, K947, K1823, K1850
 - \@topsep A1, A71, A73, A171
 - \@topsepadd A1, A59, A61, A71, A124
 - \@totalleftmargin y102,
A9, A53, A54, B240, C35, C65, C70
 - \@trivlist A48, A57, A92
 - \@tryfcolumn
. K688, K708, K726, K742, K2075
 - \@trylist K751, K754, K787, K807, K829
 - \@twoclasseserror L208, L426
 - \@twocolumnfalse K106, K154
 - \@twocolumntrue K180
 - \@twoloadclasserror L356, L421
 - \@twosidefalse K107
 - \@typein d19, d20, d27, d35
 - \@typeset@protect d79, d220,
d227, d229, l26, l32, l160, l168, s71
 - \@uclclist l852, l853, l900, O314
 - \@undefined a63, a64, a103, a104,
a105, a126, a134, a142, a149,
a200, a204, a230, a237, a297,
a298, b65, b98, b99, b114, b115,
b120, b129, b142, b177, b182,
b215, b216, b228, b238, b272,
b295, b298, b456, b499, b545,
b546, d21, d200, d278, g28, k51,
k52, k137, l145, l147, l282, l301,
m113, o391, o424, o488, q4,
q5, q6, q7, q8, q9, q10, q11,
q12, q13, q14, q15, q16, q17,
q18, q19, q20, s44, v105, D289,
 - D290, G5, G398, G399, K36,
K342, K343, L4, L346, L372,
L489, L492, L506, N2, N13,
N14, N15, N28, N30, N77, N87,
N176, N184, N192, N200, N229,
N230, N231, N232, N233, N234,
N235, N236, N237, N238, N239,
N240, N241, N242, N243, N244,
N245, N246, N247, O10, O18,
O25, O34, O53, O62, O69, O87,
O88, O201, O277, O278, O338,
O373, O374, O375, O376, O377, I33
 - \@unexpandable@noexpand d196
 - \@unexpandable@protect
d196, d232, d238, d243, k75, C225
 - \@unknownoptionerror L367, L396, L409
 - \@unprocessedoptions
. L192, L236, L343, L347, L411
 - \@unused d4, g15, g32, g59, k3, L511
 - \@unusedoptionlist
. k12, k14, L11, L145, L146, L198
 - \@upline D154, D155, D161
 - \@upordown D74, D75, D83, D104, D130
 - \@upvector D125, D161
 - \@use@ption
. L163, L175, L185, L187, L196
 - \@use@text@encoding l110, l1206
 - \@vbsphack i139
 - \@verb y136, y144
 - \@verbatim y100, y118, y121
 - \@verreq t365, t366
 - \@viipt o503
 - \@viipt o502
 - \@vipt o501
 - \@vline D59, D154
 - \@vobeyspaces y93, y118, y144
 - \@vpt o500
 - \@vspace i226
 - \@vspacer i226
 - \@vtryfc K757, K765
 - \@vvector D117, D125
 - \@warning g170
 - \@wckptelt k122, k125
 - \@whiledim f7, D36, D82
 - \@whilenoop f3
 - \@whilenum f3, C205, D31,
D184, D186, D206, D209, D406
 - \@whilesw f10, K238, K368,
K377, K415, K425, K2126, K2166
 - \@whileswnoop f10
 - \@wholewidth B113, B115, B116,
B118, B120, B121, B122, B123,
D2, D38, D40, D41, D156,
- File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- [D159](#), [D197](#), [D204](#), [D273](#), [D280](#),
[D317](#), [D323](#), [D364](#), [D365](#), [D403](#)
`\@width` [b402](#),
[d13](#), [i298](#), [l240](#), [l243](#), [p146](#),
[t522](#), [B118](#), [B120](#), [B170](#), [B177](#),
[B324](#), [B367](#), [C161](#), [C192](#), [C306](#),
[C325](#), [D106](#), [D156](#), [D159](#), [D176](#),
[D183](#), [D197](#), [D204](#), [D273](#), [D317](#),
[D403](#), [G375](#), [K1761](#), [K2115](#), [K2149](#)
`\@wrglossary` [H25](#), [H30](#)
`\@wrindex` [H8](#), [H13](#)
`\@writeckpt` [k110](#), [k119](#)
`\@writefile` [k26](#), [y43](#), [F147](#)
`\@writesetup` [K564](#)
`\@wrong@font@char` [l121](#), [o392](#), [o426](#), [o439](#)
`\@xtryfc` [K767](#), [K777](#)
`\@x@protect` [d82](#), [d219](#)
`\@x@sf` [G440](#), [G442](#)
`\@xDeclareMathDelimiter` .. [r695](#), [r751](#)
`\@xadvskip` [i139](#), [i160](#)
`\@xarg` [D56](#), [D59](#), [D64](#),
[D68](#), [D69](#), [D105](#), [D107](#), [D112](#),
[D113](#), [D117](#), [D123](#), [D131](#), [D349](#)
`\@xargarraycr` [C178](#), [C187](#), [C191](#)
`\@xargdef` [d57](#)
`\@xarraycr` [C175](#), [C176](#)
`\@xbitor` [K15](#), [K17](#)
`\@xcentercr` [y69](#), [y70](#)
`\@xdblarg` [d311](#)
`\@xdblfloat` [G268](#)
`\@xdim` [D26](#), [D32](#), [D34](#), [D353](#),
[D407](#), [D408](#), [D409](#), [D410](#), [D416](#)
`\@xeqncr` [z280](#)
`\@xexnoop` [C199](#), [C209](#)
`\@xexpast` [C200](#), [C201](#)
`\@xfloat` [G28](#), [G29](#), [G34](#), [G270](#)
`\@xfootnote` [G405](#), [G408](#)
`\@xfootnotemark` [G428](#), [G432](#)
`\@xfootnotenext` [G445](#), [G448](#)
`\@xhline` [C319](#), [C320](#)
`\@xifnch` [d297](#), [d307](#)
`\@xiipt` [o507](#), [t83](#), [t85](#), [t86](#)
`\@xipt` [o506](#), [t82](#)
`\@xivpt` [o508](#), [t84](#), [t86](#)
`\@xmpar` [G328](#), [G329](#)
`\@xnewline` [i39](#), [i40](#), [i44](#)
`\@xnext` [K10](#), [K11](#)
`\@xnthm` [E5](#), [E6](#)
`\@xobeysp` [i276](#), [y94](#), [y95](#)
`\@xprocess@ptions` .. [L154](#), [L169](#), [L181](#)
`\@xpt` [o505](#), [t81](#), [t84](#), [t85](#)
`\@xssect` [F69](#), [F70](#), [F106](#)
`\@xtabcr` [C56](#), [C57](#)
`\@xtabularcr` [C182](#), [C183](#)
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
`\@xthm` [E28](#), [E29](#)
`\@xtryfc` [K754](#), [K782](#)
`\@xtypein` [d20](#), [d22](#), [d29](#)
`\@xverbatim` [y95](#), [y118](#)
`\@xviipt` [o509](#), [t85](#), [t87](#)
`\@xxDeclareMathDelimiter` .. [r680](#), [r684](#)
`\@xxpt` [o510](#), [t86](#), [t87](#)
`\@xxvpt` [o511](#), [t87](#)
`\@xxxii` [e2](#), [l349](#), [l351](#), [G89](#),
[G158](#), [K789](#), [K790](#), [K809](#), [K810](#),
[K845](#), [K846](#), [K868](#), [K869](#), [K1913](#)
`\@xympar` [G332](#), [G336](#), [G358](#)
`\@yarg` [D56](#),
[D60](#), [D64](#), [D65](#), [D74](#), [D112](#),
[D118](#), [D125](#), [D127](#), [D154](#), [D349](#)
`\@yargarraycr` [C179](#), [C189](#), [C193](#)
`\@yargd@f` [d84](#)
`\@yargdef` [d61](#), [d71](#), [d84](#), [d100](#)
`\@ydim` [D27](#), [D32](#), [D34](#), [D354](#),
[D411](#), [D412](#), [D413](#), [D414](#), [D415](#)
`\@yeqncr` [z280](#)
`\@ympar` [G328](#), [G333](#)
`\@ynthm` [E5](#), [E14](#)
`\@ythm` [E28](#), [E29](#)
`\@ytryfc` [K800](#), [K819](#), [K823](#)
`\@yyarg` [D64](#), [D65](#), [D66](#), [D69](#), [D131](#), [D349](#)
`\@ztryfc` [K828](#), [K839](#)
`\[` [o346](#), [z184](#), [z240](#), [z321](#), [O260](#)
`\ \` [66](#), [a45](#), [a46](#), [a69](#),
[a242](#), [a243](#), [a244](#), [a245](#), [a248](#),
[a255](#), [a256](#), [a257](#), [a258](#), [a261](#),
[a268](#), [a269](#), [a270](#), [a271](#), [a274](#),
[a281](#), [a283](#), [a284](#), [a287](#), [a290](#),
[b13](#), [d195](#), [d313](#), [g231](#), [i35](#), [i309](#),
[k204](#), [k219](#), [l447](#), [o334](#), [t170](#),
[y76](#), [y83](#), [y89](#), [y97](#), [z262](#), [z381](#),
[B245](#), [B351](#), [B353](#), [C62](#), [C143](#),
[C153](#), [C167](#), [D49](#), [N387](#), [O245](#)
`\{` [a3](#), [a7](#), [a69](#), [b2](#), [b13](#), [g22](#), [l257](#), [l449](#),
[o335](#), [t168](#), [y96](#), [z59](#), [z108](#), [O248](#)
`\}` [a8](#), [a69](#), [b3](#), [b13](#), [g21](#), [l258](#),
[l450](#), [o336](#), [t169](#), [y96](#), [z59](#), [O249](#)
`\]` .. [b448](#), [o347](#), [z184](#), [z241](#), [z345](#), [O261](#)
`\^` ... [a58](#), [a67](#), [a70](#), [a114](#), [a301](#), [b7](#),
[b9](#), [b11](#), [b14](#), [b367](#), [b368](#), [b382](#),
[b383](#), [d5](#), [d314](#), [i309](#), [i311](#), [i313](#),
[l181](#), [l236](#), [l309](#), [l380](#), [l388](#), [l445](#),
[l528](#), [l535](#), [l539](#), [l544](#), [l549](#), [l554](#),
[l561](#), [l567](#), [l568](#), [l574](#), [l579](#), [l634](#),
[o332](#), [o333](#), [o338](#), [L434](#), [L435](#),
[L436](#), [L488](#), [L491](#), [L494](#), [O186](#),
[O187](#), [O188](#), [O189](#), [O191](#), [O192](#),
[O193](#), [O194](#), [O196](#), [O246](#), [O252](#),
[O253](#), [O254](#), [O255](#), [O268](#), [O269](#),

O270, O302, O303, O304, O305,
 O307, O308, O309, O310, O312
 _ a70, b8, b14,
 d314, l263, t173, z166, z167, O247
 \^ l182,
 l310, l344, l378, l386, l464, l526,
 l533, l537, l542, l547, l552, l559,
 l563, l564, l572, l577, l635, l674,
 o348, s168, y145, B236, C61, O262
 \| l448, m78, m89, t479, O263
 \~ a70, b10, b14,
 d314, g20, i278, l189, l237, l311,
 l389, l446, l529, l541, l545, l555,
 l571, l575, l636, y139, y149, O250

 \sq a69, a86, b13,
 b367, b385, d313, g19, g20, g21,
 g22, g25, i277, o331, o497, t171,
 y93, y94, E36, E38, L111, O240, I17

A

\A O181, O265, O297
 \a [l173](#), [C1](#), O170, O266, O286
 \AA [b373](#), l190, l352, l416
 \aa [b373](#), l195, l346, l426
 \abovedisplayshortskip .. b348, z389
 \abovedisplayskip b347,
 z382, z384, z386, z387, z388, z389
 \accent l71, l320, l347, l401, l647
 \accent@spacefactor l70, l71, l72
 \active a59, a114, a301, b10,
 b11, b382, b383, b385, y93, y94,
 y138, y147, z151, z166, K558,
 L434, L435, L436, L488, L491, L494
 \active@math@prime .. [z150](#), z151, K562
 \acute t424
 \add@accent [l65](#), [l67](#)
 \add\to\callback [476](#), [N597](#)
 \addcontentsline [F53](#), [F63](#), [F142](#), [G16](#)
 \addpenalty [i166](#), [A124](#), [A170](#),
 [A175](#), [F33](#), [K312](#), [K1063](#), [K1229](#)
 \addto@hook
 . [o117](#), [o119](#), [o499](#), r263, r359,
 r363, r380, r504, r510, r518,
 r534, r537, r540, r881, r888, r891
 \addtocontents [F143](#), [F144](#)
 \addtocounter [132](#), [m6](#), m18
 \addtolength [138](#), [n16](#), z384, z386
 \addtoversion [q20](#), [q139](#)
 \addvspace [i153](#), y70, [A124](#),
 [A171](#), [A172](#), [A176](#), [A224](#), [F33](#)
 \adjdemerits b327
 \AE [l191](#), [l325](#), [l417](#), [l652](#), O327
 File Key: a=lt`dirchk.dtx`, b=lt`plain.dtx`, c=lt`vers.dtx`, d=lt`defs.dtx`,
 e=lt`alloc.dtx`, f=lt`cntrl.dtx`, g=lt`error.dtx`, h=lt`par.dtx`, i=lt`space.dtx`,
 j=lt`logos.dtx`, k=lt`files.dtx`, l=lt`outenc.dtx`, m=lt`counts.dtx`, n=lt`length.dtx`,
 o=lt`fssbas.dtx`, p=lt`fssstrc.dtx`, q=lt`fsscmp.dtx`, r=lt`fssdcl.dtx`, s=lt`fssini.dtx`,
 t=lt`fontdef.dtx`, u=lt`preload.dtx`, v=lt`fontcmd.dtx`, w=lt`pageno.dtx`, x=lt`xref.dtx`,
 y=lt`miscen.dtx`, z=lt`math.dtx`, A=lt`lists.dtx`, B=lt`boxes.dtx`, C=lt`tab.dtx`,
 D=lt`pictur.dtx`, E=lt`thm.dtx`, F=lt`ssect.dtx`, G=lt`float.dtx`, H=lt`tidxglo.dtx`,
 I=lt`bibl.dtx`, J=lt`page.dtx`, K=lt`output.dtx`, L=lt`class.dtx`, M=lt`hyphen.dtx`,
 N=lt`luatex.dtx`, O=lt`final.dtx`

\ae [l196](#), [l328](#), [l427](#), [l656](#), O327
 \afterassignment b398, b401,
 d233, d239, l162, l170, o262, z129
 \aftergroup o56, o276,
 p156, p222, r114, r121, r129,
 v47, y142, B101, K565, K573, K574
 \aleph t227
 \alloc@ b83, b84,
 b85, b86, b87, b88, b89, b90,
 b91, b92, [b219](#), o15, N20, N24, N38
 \allocationnumber
 [b37](#), b57, b69, b136,
 b137, b138, b188, b189, b221,
 b222, b223, b234, b235, b236,
 b253, b258, b264, b265, b278,
 b279, b280, C4, C9, N52, N53,
 N54, N93, N207, O38, O39, O40
 \allowbreak [b405](#), z40
 \Alph [132](#), [m49](#)
 \alph [132](#), [m48](#)
 \alpha t187
 \alpha@elt
 . [r45](#), r267, r454, r556, r880, r881
 \alpha@list r41, [r43](#), r276, r442, r454,
 r499, r554, r555, r876, r882, r883
 \amalg t293
 \and [344](#), [F14](#)
 \angle t243
 \approx t333
 \arabic [132](#), [m45](#), E33
 \arccos z13
 \arcsin z10
 \arctan z16
 \arg z26
 \array [C141](#)
 \arraycolsep
 z265, z266, z394, z395, C219, [C297](#)
 \arrayrulewidth
 C283, [C297](#), C305, C306,
 C318, C322, C325, C335, C337
 \arraystretch C159, C160, [C301](#)
 \Arrowvert t475
 \arrowvert t473
 \ast t151, t309
 \asympt t357
 \AtBeginDocument . k47, [L381](#), I34, I48
 \AtBeginDvi [K93](#)
 \AtEndDocument y9, [L381](#)
 \AtEndOfClass z320, [L381](#)
 \AtEndOfPackage ... [L192](#), [L236](#), [L381](#)
 \atopwithdelims z57, z58, z59
 \attribute N81
 \attributedef N81, N213
 \attributezero N213

- `\author` 344, [F5](#)
B
`\b` 1183, 1316, 1397, 1643
`\backslash` t170, t494
`\badness` b306
`\bar` t428
`\baselineskip` b366, b396,
 b432, p140, p141, p142, p144,
 p145, t418, z112, z113, z121,
 z127, z131, B243, C171, D46,
 D166, K216, K247, K593, K608
`\baselinestretch`
 o253, p118, p119, p138, p199
`\batchmode`
 k183, k184, q106, s135, O347, O368
`\begin` g201, g203, l600, p7, t4,
 u4, y51, y52, z325, z337, F14,
 F17, G256, G258, K70, L245, M3
`\belowdisplayskip` .. b350, z388
`\belowdisplayshortskip` b349, z387
`\best@size` p392, p416, p422, p428
`\beta` t188
`\bezier` 318, D367, [D368](#)
`\bfdefault` s15, [t32](#)
`\bfseries`
 . s13, s14, v19, x13, E36, E38, I20
`\bgroup` [b380](#)
`\biblecite` I7, I9, [I10](#)
`\bibdata` [I25](#), [I29](#)
`\bibitem` [I3](#)
`\bibliography` 376, [I27](#)
`\bibliographystyle` 376, [I32](#)
`\bibstyle` [I25](#), [I37](#)
`\Big` t525, z44, z45, z46
`\big` t524, [z41](#)
`\bigbreak` [b412](#)
`\bigcap` t263
`\bigcirc` t306
`\bigcup` t264
`\Bigg` t527, z50, z51, z52
`\bigg` t526, z47, z48, z49
`\Biggl` z50
`\biggl` z47
`\Biggm` z51
`\biggm` z48
`\Biggr` z52
`\biggr` z49
`\Bigl` z44
`\bigl` z41
`\Bigm` z45
`\bigm` z42
`\bigodot` t271
`\bigoplus` t270
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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx
`\bigotimes` t269
`\Bigr` z46
`\bigr` z43
`\bigskip` b417, [i256](#)
`\bigskipamount` . b416, i258, [i259](#), G371
`\bigsqcup` t274
`\bigtriangledown` t279, t280
`\bigtriangleup` t278, t281
`\biguplus` t262
`\bigvee` t260
`\bigwedge` t261
`\binoppenalty` b318
`\bm@b` [B36](#)
`\bm@c` [B36](#)
`\bm@l` [B36](#)
`\bm@r` [B36](#)
`\bm@s` [B36](#)
`\bm@t` [B36](#)
`\bmod` z35
`\boldmath` j14, s63
`\bordermatrix` z115
`\bot` t242
`\botfigrule` K652, [K2214](#)
`\botmark` J36, K618
`\bottomfraction` G279, [K2183](#)
`\bowtie` t390
`\Box` s106
`\boxmaxdepth`
 b341, D246, D291, D330, K460,
 K480, K520, K625, K634, K674
`\brace` z59
`\braced` .. t459, t463, t464, t466, t468
`\bracelu` t461, t465, t467
`\bracerd` t460, t465, t467
`\braceru` t462, t464, t468
`\bracevert` [t512](#)
`\brack` z58
`\break` [b405](#), b410, i53
`\breve` t429
`\brokenpenalty` b323
`\buildrel` t377, [z107](#)
`\bullet` t295
`\bx@A` K30, K57
`\bx@AA` K40
`\bx@B` K30, K57
`\bx@BB` K40
`\bx@C` K30, K57
`\bx@CC` K40
`\bx@D` K30, K57
`\bx@DD` K40
`\bx@E` K30, K57
`\bx@EE` K40
`\bx@F` K31, K58
`\bx@FF` K41

`\bx@G` K31, K58
`\bx@GG` K41
`\bx@H` K31, K58
`\bx@HH` K41
`\bx@I` K31, K58
`\bx@II` K41
`\bx@J` K31, K58
`\bx@JJ` K41
`\bx@K` K32, K59
`\bx@KK` K42
`\bx@L` K32, K59
`\bx@LL` K42
`\bx@M` K32, K59
`\bx@MM` K42
`\bx@N` K32, K59
`\bx@NN` K42
`\bx@O` K33, K60
`\bx@OO` K43
`\bx@P` K33, K60
`\bx@PP` K43
`\bx@Q` K33, K60
`\bx@QQ` K43
`\bx@R` K33, K60
`\bx@RR` K43
`\bx@S` K38
`\bx@SS` K44
`\bx@T` K38
`\bx@TT` K44
`\bx@U` K38
`\bx@UU` K44
`\bx@V` K38
`\bx@VV` K44
`\bx@W` K39
`\bx@WW` K45
`\bx@X` K39
`\bx@XX` K45
`\bx@Y` K39
`\bx@YY` K45
`\bx@Z` K39
`\bx@ZZ` K25, K45, K55

C

`\c` 1184, 1319, 1382,
1400, 1490, 1492, 1517, 1519, 1532,
1558, 1584, 1585, 1586, 1587, 1588,
1589, 1590, 1591, 1592, 1593, 1646
`\c@bottomnumber` .. G273, G278, K2181
`\c@dbltopnumber`
..... G272, G287, G301, K2188
`\c@enumi` A227
`\c@enumii` A227, A227
`\c@enumiv` A227
`\c@equation` z242, z275, z401
`\c@errorcontextlines` g163
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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

`\c@footnote` F11, G377, G434
`\c@mpfootnote` B271, G379
`\c@ncel` t369, t370
`\c@page` w3, w6, w7, K145, K1728
`\c@secnumdepth` .. F39, F54, F64, F123
`\c@tocdepth` F123, F150
`\c@topnumber` G271, G275, K2177
`\c@totalnumber` ... G274, G280, K2184
`\cal` s169
`\calculate@math@sizes` ... o475, p173
`\call_callback` 476, N575
`\callback.register` N504
`\callback_descriptions` .. 476, N706
`\cap` t286
`\capitalacute` 1720, 11008
`\capitalbreve` 1727, 11015
`\capitalcaron` 1726, 11014
`\capitalcedilla` 1713, 11005
`\capitalcircumflex` 1721, 11009
`\capitaldieresis` 1723, 11011
`\capitaldotaccent` 1729, 11017
`\capitalgrave` 1719, 11007
`\capitalhungarumlaut` 1724, 11012
`\capitalmacron` 1728, 11016
`\capitalnewtie` 1733, 11082, 11083
`\capitalogonek` 1716, 11006
`\capitalring` 1725, 11013
`\capitaltie` 1731, 11078, 11079
`\capitaltilde` 1722, 11010
`\caption` G4
`\cases` z108
`\catcodetable` N91, N112
`\catcodetable@atletter` 474, N96, N238
`\catcodetable@initex` . 474, N96, N235
`\catcodetable@latex` .. 474, N96, N237
`\catcodetable@string` . 474, N96, N236
`\cdot` t308
`\cdotp` t410, t416
`\cdots` t416
`\c@elt`
o61, o81, o92, o93, o114, o117,
o119, r201, r283, r338, r402, r483
`\c@p@list` o63, o79, o93, o121,
o122, r219, r285, r340, r404, r485
`\center` y73
`center` (environment) y73
`\centering` y73, y75
`\centerline` B368
`\cf@encoding` 134, 141, 144,
151, 1114, o221, o231, o241, o260
`\ch@ck` b199,
b200, b201, b202, b220, b230,
b231, b232, b233, b260, b262,
b274, b275, b276, b277, b283, L445

- `\changes` [l597](#),
[G257](#), [K71](#), [N225](#), [N226](#), [O214](#)
 - `\char` ... [l318](#), [l321](#), [l354](#), [l357](#), [l368](#),
[l375](#), [l399](#), [l403](#), [l408](#), [l411](#), [l413](#),
[l415](#), [l617](#), [l645](#), [l648](#), [l678](#), [l685](#),
[l692](#), [l715](#), [l718](#), [l766](#), [s69](#), [y150](#),
[z148](#), [D111](#), [D139](#), [D153](#), [D161](#),
[D164](#), [D233](#), [D271](#), [D276](#), [D315](#),
[D319](#), [D334](#), [D335](#), [D337](#), [D348](#)
 - `\chardef` [a59](#), [a65](#),
[a66](#), [b10](#), [b16](#), [b17](#), [b18](#), [b19](#),
[b20](#), [b58](#), [b64](#), [b66](#), [b70](#), [b76](#),
[b77](#), [b87](#), [b89](#), [b90](#), [b91](#), [b92](#),
[b101](#), [b107](#), [b108](#), [b121](#), [b123](#),
[b187](#), [b235](#), [b239](#), [b241](#), [b264](#),
[b279](#), [b426](#), [b427](#), [b428](#), [e2](#), [l18](#),
[o15](#), [C4](#), [C9](#), [L444](#), [N20](#), [N24](#),
[N38](#), [N47](#), [N48](#), [N91](#), [N160](#),
[N214](#), [O28](#), [O47](#), [O98](#), [O99](#),
[O100](#), [O101](#), [O102](#), [O103](#), [O104](#)
 - `\chardef@text@cmd` [l3](#)
 - `\charzero` [N214](#)
 - `\check` [t430](#)
 - `\check@command` [d164](#), [d166](#)
 - `\check@icl`
.. [v9](#), [v27](#), [v32](#), [v38](#), [v46](#), [v53](#), [v55](#)
 - `\check@icr`
.. [v9](#), [v27](#), [v33](#), [v39](#), [v47](#), [v56](#), [v61](#)
 - `\check@mathfonts` [j5](#),
[l251](#), [l277](#), [l291](#), [o282](#), [o284](#), [p204](#)
 - `\check@nocorr@` [v29](#)
 - `\check@range` [p333](#), [p334](#)
 - `\check@single` [p332](#), [p354](#)
 - `\CheckCommand` [d164](#)
 - `\CheckEncodingSubset` . [l946](#), [l1003](#),
[l1004](#), [l1072](#), [l1189](#), [l1192](#), [l1206](#)
 - `\chi` [t207](#)
 - `\choose` [z57](#)
 - `\circ` [t305](#)
 - `\circle` [D235](#), [D328](#)
 - `\citation` [l11](#), [l19](#), [l43](#)
 - `\cite` [376](#), [l12](#)
 - `\cl@ckpt` [k122](#), [m35](#)
 - `\cl@page` [w4](#)
 - `\ClassError` [g84](#)
 - `\ClassInfo` [g84](#)
 - `\ClassWarning` [g84](#)
 - `\ClassWarningNoLine` [g84](#)
 - `\cleaders` [b446](#), [t454](#), [t457](#)
 - `\cleardoublepage` [K145](#)
 - `\clearpage` ... [k91](#), [k109](#), [y12](#), [y49](#),
[K132](#), [K145](#), [K150](#), [K174](#), [K381](#),
[K384](#), [K388](#), [K429](#), [K435](#), [K2079](#)
 - `\cline` [C326](#)
 - `\clubpenalty` . [b320](#), [k10](#), [k19](#), [A128](#),
[A194](#), [A196](#), [F83](#), [F89](#), [F113](#), [F118](#)
 - `\clubsuit` [t255](#)
 - `\col@number` .. [K102](#), [K155](#), [K182](#), [K194](#)
 - `\colon` [t411](#)
 - `\color@begingroup` [o493](#),
[z87](#), [z103](#), [B28](#), [B62](#), [B129](#),
[B267](#), [B300](#), [C47](#), [C51](#), [G423](#), [K465](#)
 - `\color@endbox` [B62](#),
[G253](#), [G348](#), [K198](#), [K602](#), [K612](#)
 - `\color@endgroup` .. [o498](#), [z87](#), [z103](#),
[B28](#), [B62](#), [B87](#), [B108](#), [B131](#),
[B287](#), [B303](#), [C49](#), [G426](#), [K469](#)
 - `\color@hbox` [B62](#), [K599](#), [K609](#)
 - `\color@setgroup` [B62](#), [B87](#), [B106](#)
 - `\color@vbox` [B62](#),
[G96](#), [G165](#), [G339](#), [G361](#), [K189](#)
 - `\columnsep` [k21](#), [K88](#), [K176](#)
 - `\columnseprule` ... [K89](#), [K2115](#), [K2149](#)
 - `\columnwidth`
[k18](#), [k21](#), [k22](#), [k24](#), [B269](#), [B296](#),
[G99](#), [G168](#), [G419](#), [K87](#), [K151](#),
[K152](#), [K153](#), [K175](#), [K176](#), [K177](#),
[K178](#), [K179](#), [K1753](#), [K1755](#),
[K2113](#), [K2117](#), [K2145](#), [K2151](#)
 - `\cong` [t365](#)
 - `\contentsline` [F143](#), [F148](#)
 - `\coprod` [t259](#)
 - `\copyright` [l235](#), [l265](#), [s89](#)
 - `\cos` [z12](#)
 - `\cosh` [z14](#)
 - `\cot` [z18](#)
 - `\coth` [z19](#)
 - `\count@` [a61](#),
[a174](#), [a175](#), [a176](#), [a181](#), [b41](#),
[b184](#), [b185](#), [b190](#), [b192](#), [b198](#),
[b199](#), [b200](#), [b201](#), [b202](#), [b203](#),
[b402](#), [b403](#), [c13](#), [c14](#), [c15](#), [c16](#),
[c17](#), [c19](#), [d146](#), [d150](#), [p22](#), [p256](#),
[p258](#), [p280](#), [p281](#), [r260](#), [r262](#),
[r266](#), [r581](#), [r582](#), [r583](#), [r624](#),
[r625](#), [r626](#), [r668](#), [r669](#), [r670](#),
[r708](#), [r709](#), [r710](#), [r716](#), [r717](#),
[r718](#), [r762](#), [r763](#), [r764](#), [r770](#),
[r771](#), [r772](#), [r813](#), [r814](#), [r815](#),
[r821](#), [r822](#), [r823](#), [v98](#), [v101](#),
[D405](#), [D406](#), [D407](#), [D410](#), [D411](#),
[D414](#), [D418](#), [O166](#), [O167](#), [O176](#),
[O178](#), [O282](#), [O283](#), [O292](#), [O294](#)
 - `\countdef` . [a61](#), [b37](#), [b38](#), [b39](#), [b41](#),
[b51](#), [b83](#), [w3](#), [N78](#), [N88](#), [N177](#),
[N185](#), [N193](#), [N201](#), [N215](#), [O55](#)
 - `\CountZero` [N215](#)
- File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

<code>\cr</code>	b376, 1995, 1999, z118, z122, z270, z300, z398, C165, C176, C183, C192, C193, C336, D51, D53, D54	<code>\cyrchrdsc</code>	1856
<code>\crrc</code>	b433, l276, l293, l294, l317, l321, l324, l398, l402, l406, l408, l411, l616, l644, l648, l651, l715, l718, l765, l1000, s91, t243, t244, t246, t367, t370, t374, t438, t439, t440, t441, t442, t443, t444, t445, t446, t447, t448, t450, z109, z111, z112, z113, z118, z120, z121, z122, z140, z141, C144, C145, D51	<code>\CYRCHVCRS</code>	1857
<code>\create_callback</code>	476, N556	<code>\cyrchvcrs</code>	1857
<code>\cs</code>	1597, K72	<code>\CYRD</code>	1857
<code>\csc</code>	z21	<code>\cyrd</code>	1857
<code>\cup</code>	t287	<code>\CYRDELTA</code>	1857
<code>\curr@fontshape</code>	l130, o53, o297, o305, o309, o311, o374, o380, o383, o392, o399, o401, o409, o415, o418, o426, o433, o435, p92, p100, p121, p431, p451, p483, p496, r223, r228	<code>\cyrdelta</code>	1857
<code>\curr@math@size</code>	o286, p210, p216, p221, p238	<code>\CYRDJE</code>	1858
<code>\CurrentOption</code>	l882, l884, l896, L13, L147, L157, L158, L159, L164, L171, L172, L173, L176, L183, L184, L188, L189, L190, L197, L199, L203, L204, L205, L316, L398, L399, L408, L409	<code>\cyrdje</code>	1858
<code>\CYRA</code>	1854	<code>\CYRDZE</code>	1858
<code>\cyra</code>	1854, 1899	<code>\cyrdze</code>	1858
<code>\CYRABHCH</code>	1854	<code>\CYRDZHE</code>	1858
<code>\cyrabhch</code>	1854	<code>\cyrdzhe</code>	1858
<code>\CYRABHCHDSC</code>	1854	<code>\CYRE</code>	1858
<code>\cyrabhchdsc</code>	1854	<code>\cyre</code>	1858
<code>\CYRABHDZE</code>	1855	<code>\CYREPS</code>	1859
<code>\cyrabhdze</code>	1854	<code>\cyreps</code>	1858
<code>\CYRABHHA</code>	1855	<code>\CYREREV</code>	1859
<code>\cyrabhha</code>	1855	<code>\cyrerev</code>	1859
<code>\CYRAE</code>	1855	<code>\CYRERY</code>	1859
<code>\cyrae</code>	1855	<code>\cyrery</code>	1859
<code>\CYRB</code>	1855	<code>\CYRF</code>	1859
<code>\cyrb</code>	1855	<code>\cyrf</code>	1859
<code>\CYRBYUS</code>	1856	<code>\CYRFITA</code>	1860
<code>\cyrbyus</code>	1855	<code>\cyrfita</code>	1859
<code>\CYRC</code>	1856	<code>\CYRG</code>	1860
<code>\cyrcc</code>	1856	<code>\cyrg</code>	1860
<code>\CYRCH</code>	1856	<code>\CYRGDSC</code>	1860
<code>\cyrch</code>	1856	<code>\cyrgdsc</code>	1860
<code>\CYRCHLDSC</code>	1856	<code>\CYRGDSCHCRS</code>	1860
<code>\cyrchldsc</code>	1856	<code>\cyrgdschcrs</code>	1860
<code>\CYRCHRDC</code>	1857	<code>\CYRGHCRS</code>	1861
<code>\File Key:</code>	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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx	<code>\CYRGHCRS</code>	1861
		<code>\cyrghcrs</code>	1861
		<code>\CYRGHK</code>	1861
		<code>\cyrghk</code>	1861
		<code>\CYRGUP</code>	1861
		<code>\cyrgup</code>	1861
		<code>\CYRH</code>	1861
		<code>\cyrh</code>	1861
		<code>\CYRHDSC</code>	1862
		<code>\cyrhdsc</code>	1862
		<code>\CYRHHCRCR</code>	1862
		<code>\cyrhhcrs</code>	1862
		<code>\CYRHHK</code>	1862
		<code>\cyrhkh</code>	1862
		<code>\CYRHRDSN</code>	1863
		<code>\cyrhrdsn</code>	1862
		<code>\CYRI</code>	1863
		<code>\cyri</code>	1863
		<code>\CYRIE</code>	1863
		<code>\cyrie</code>	1863
		<code>\CYRII</code>	1863
		<code>\cyrii</code>	1863
		<code>\CYRISHRT</code>	1863

<code>\cyrishrt</code>	1863	<code>\cyrq</code>	1870
<code>\CYRISHRTDSC</code>	1864	<code>\CYRR</code>	1870
<code>\cyrishrtdsc</code>	1864	<code>\cyrr</code>	1870
<code>\CYRIZH</code>	1864	<code>\CYRRDSC</code>	1870
<code>\cyrizh</code>	1864	<code>\cyrrdsc</code>	1870
<code>\CYRJE</code>	1864	<code>\CYRRHK</code>	1871
<code>\cyrje</code>	1864	<code>\cyrrhk</code>	1870
<code>\CYRK</code>	1864	<code>\CYRRTICK</code>	1871
<code>\cyrk</code>	1864	<code>\cyrrtick</code>	1871
<code>\CYRKBEAK</code>	1865	<code>\CYRS</code>	1871
<code>\cyrkbeak</code>	1865	<code>\cyrs</code>	1871
<code>\CYRKDSC</code>	1865	<code>\CYRSACRS</code>	1871
<code>\cyrkdsc</code>	1865	<code>\cyrsacrs</code>	1871
<code>\CYRKHCRS</code>	1865	<code>\CYRSCHWA</code>	1872
<code>\cyrkhcrs</code>	1865	<code>\cyrschwa</code>	1872
<code>\CYRKHK</code>	1866	<code>\CYRSDSC</code>	1872
<code>\cyrkhk</code>	1865	<code>\cyrsdsc</code>	1872
<code>\CYRKVCRS</code>	1866	<code>\CYRSEMISFTSN</code>	1872
<code>\cyrkvcrs</code>	1866	<code>\cyrsemisftsn</code>	1872
<code>\CYRL</code>	1866	<code>\CYRSFTSN</code>	1873
<code>\cyrl</code>	1866	<code>\cyrsftsn</code>	1873
<code>\CYRLDSC</code>	1866	<code>\CYRSH</code>	1873
<code>\cyrl dsc</code>	1866	<code>\cyrsh</code>	1873
<code>\CYRLHK</code>	1867	<code>\CYRSHCH</code>	1873
<code>\cyrlhk</code>	1866	<code>\cyrshch</code>	1873
<code>\CYRLJE</code>	1867	<code>\CYRSHHA</code>	1873
<code>\cyrlje</code>	1867	<code>\cyrshha</code>	1873
<code>\CYRM</code>	1867	<code>\CYRT</code>	1874
<code>\cym</code>	1867	<code>\cyrt</code>	1874
<code>\CYRMDSC</code>	1867	<code>\CYRTDSC</code>	1874
<code>\cyrmdsc</code>	1867	<code>\cyrt dsc</code>	1874
<code>\CYRMHK</code>	1867	<code>\CYRTETSE</code>	1874
<code>\cymhk</code>	1867	<code>\cyrtetse</code>	1874
<code>\CYRN</code>	1868	<code>\CYRTSHE</code>	1874
<code>\cyrn</code>	1868	<code>\cyrtshe</code>	1874
<code>\CYRNDSC</code>	1868	<code>\CYRU</code>	1875
<code>\cyrndsc</code>	1868	<code>\cyru</code>	1875
<code>\CYRNG</code>	1868	<code>\CYRUSHRT</code>	1875
<code>\cyrng</code>	1868	<code>\cyrushrt</code>	1875
<code>\CYRNHK</code>	1868	<code>\CYRV</code>	1875
<code>\cyrnhk</code>	1868	<code>\cyrv</code>	1875
<code>\CYRNJE</code>	1869	<code>\CYRW</code>	1875
<code>\cyrnje</code>	1868	<code>\cyrw</code>	1875
<code>\CYRNLHK</code>	1869	<code>\CYRY</code>	1875
<code>\cyrnlhk</code>	1869	<code>\cyry</code>	1875
<code>\CYRO</code>	1869	<code>\CYRYA</code>	1876
<code>\cyro</code>	1869	<code>\cyrya</code>	1876
<code>\CYROTLD</code>	1869	<code>\CYRYAT</code>	1876
<code>\cyrotld</code>	1869	<code>\cyryat</code>	1876
<code>\CYRP</code>	1869	<code>\CYRYHCRS</code>	1876
<code>\cyrp</code>	1869	<code>\cyryhcrs</code>	1876
<code>\CYRPHK</code>	1870	<code>\CYRYI</code>	1876
<code>\cyrphk</code>	1870	<code>\cyryi</code>	1876
<code>\CYRQ</code>	1870	<code>\CYRYO</code>	1877

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 fsstrc.dtx, q=lt fsscmp.dtx, r=lt fssdcl.dtx, s=lt fssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=lt fntcmd.dtx, w=lt pageno.dtx, x=lt xref.dtx,
y=lt miscen.dtx, z=lt math.dtx, A=lt lists.dtx, B=lt boxes.dtx, C=lt tab.dtx,
D=lt pictur.dtx, E=lt thm.dtx, F=lt sect.dtx, G=lt float.dtx, H=lt idxglo.dtx,
I=lt bibl.dtx, J=lt page.dtx, K=lt output.dtx, L=lt class.dtx, M=lt hyphen.dtx,
N=lt luatex.dtx, O=lt final.dtx

- `\cyrzo` 1876
`\CYRYU` 1877
`\cyrzu` 1877
`\CYRZ` 1877
`\cyrz` 1877
`\CYRZDSC` 1877
`\cyrzdsc` 1877
`\CYRZH` 1877
`\cyrzh` 1877
`\CYRZHDSC` 1878
`\cyrzhdsc` 1878
- D**
- `\d` 1185, 1322, 1404, 1649
`\dag` 1261
`\dagger` 1261, m74, m80, m88, m89, t289
`\dashbox` D166
`\dashv` t317
`\date` 344, F7
`\day` a180, L462
`\dblfigrule` K677, K2214
`\dblfloatpagefraction`
 G291, G305, K2191
`\dblfloatsep`
 K663, K675, K1550, K1676, K2198
`\dbltextfloatsep` K196,
 K204, K679, K1549, K1675, K2198
`\dbltopfraction` .. G288, G302, K2190
`\ddag` 1262
`\ddagger` 1262, m75, m81, m88, m90, t288
`\ddot` t426
`\ddots` t421
`\deadcycles` k115, y39, y49, K273
`\declare@robustcommand` d198
`\DeclareEncodingSubset` 1930,
 1937, 1938, 1939, 1940, 11216,
 11217, 11218, 11219, 11220, 11221,
 11222, 11223, 11224, 11225, 11226,
 11227, 11228, 11229, 11230, 11231,
 11232, 11233, 11234, 11235, 11236,
 11237, 11238, 11239, 11240, 11241,
 11242, 11243, 11244, 11245, 11246,
 11247, 11248, 11249, 11250, 11251,
 11252, 11253, 11254, 11255, 11256,
 11257, 11258, 11259, 11260, 11261,
 11262, 11263, 11264, 11265, 11266,
 11267, 11268, 11269, 11270, 11271,
 11272, 11273, 11274, 11275, 11276
`\DeclareErrorFont` o352, r232, s115, t28
`\DeclareFixedFont` o40
`\DeclareFontEncoding`
 1304, 1385, 1596, 1622,
 1628, 1711, o83, t45, t46, t47, t48
`\DeclareFontEncoding@` . o87, o89, o104
`\DeclareFontEncodingDefaults` ...
 o133, q90, q91, t15
`\DeclareFontFamily` o58, q85, q86
`\DeclareFontShape`
 o19, o21, q25, q27, q81, q82
`\DeclareFontShape@` o22, o23
`\DeclareFontSubstitution`
 1629, 1712,
 o106, t16, t17, t49, t50, t51, t52
`\DeclareMathAccent`
 r573, r612, t424, t425,
 t426, t427, t428, t429, t430,
 t431, t432, t433, t434, t435, t436
`\DeclareMathAlphabet`
 q119, q123, q125, q132,
 r399, r562, r573, t70, t71, t72, t73
`\DeclareMathAlphabetCharacter` . r674
`\DeclareMathDelimiter`
 r676, t174, t175,
 t176, t177, t178, t179, t182,
 t184, t185, t469, t471, t473,
 t475, t477, t480, t482, t484,
 t486, t488, t490, t492, t494,
 t496, t498, t500, t502, t504,
 t506, t508, t510, t512, t514, t516
`\DeclareMathRadical` r793, t437
`\DeclareMathSizes` o170,
 o176, o198, t76, t77, t78, t79,
 t80, t81, t82, t83, t84, t85, t86, t87
`\DeclareMathSizes*` o170
`\DeclareMathSymbol`
 r616, r675, r692, t88,
 t89, t90, t91, t92, t93, t94, t95,
 t96, t97, t98, t99, t100, t101,
 t102, t103, t104, t105, t106,
 t107, t108, t109, t110, t111,
 t112, t113, t114, t115, t116,
 t117, t118, t119, t120, t121,
 t122, t123, t124, t125, t126,
 t127, t128, t129, t130, t131,
 t132, t133, t134, t135, t136,
 t137, t138, t139, t140, t141,
 t142, t143, t144, t145, t146,
 t147, t148, t149, t150, t151,
 t152, t153, t154, t155, t156,
 t157, t158, t159, t160, t161,
 t162, t163, t164, t165, t166,
 t167, t168, t169, t170, t180,
 t181, t183, t187, t188, t189,
 t190, t191, t192, t193, t194,
 t195, t196, t197, t198, t199,
 t200, t201, t202, t203, t204,
 t205, t206, t207, t208, t209,
 t210, t211, t212, t213, t214,
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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- t215, t216, t217, t218, t219,
t220, t221, t222, t223, t224,
t225, t226, t227, t229, t230,
t231, t232, t233, t234, t235,
t236, t237, t238, t239, t241,
t242, t247, t248, t249, t250,
t252, t253, t254, t255, t256,
t257, t258, t259, t260, t261,
t262, t263, t264, t265, t267,
t268, t269, t270, t271, t272,
t274, t275, t276, t277, t278,
t279, t282, t284, t286, t287,
t288, t289, t290, t291, t292,
t293, t294, t295, t296, t297,
t298, t299, t300, t301, t302,
t303, t304, t305, t306, t307,
t308, t309, t310, t311, t312,
t313, t314, t315, t316, t317,
t318, t319, t320, t321, t322,
t323, t324, t325, t327, t329,
t331, t332, t333, t334, t335,
t336, t337, t338, t339, t340,
t341, t343, t344, t345, t346,
t347, t349, t351, t353, t354,
t355, t356, t357, t358, t359,
t360, t361, t362, t363, t385,
t387, t409, t410, t411, t459,
t460, t461, t462, t518, t519, t520
- `\DeclareMathVersion` [r245](#), s2, s3
- `\DeclareOldFontCommand` . . . [v108](#), [v124](#)
- `\DeclareOption` . . . [455](#), [l881](#), [l937](#),
[l938](#), [l939](#), [l940](#), [l941](#), [l943](#), [p29](#),
[p37](#), [p45](#), [p53](#), [p56](#), [p60](#), [L130](#), [L419](#)
- `\DeclareOption*` [455](#), [L130](#)
- `\DeclarePreloadSizes`
. [o150](#), [q95](#), [q96](#), [u19](#), [u21](#),
[u22](#), [u23](#), [u25](#), [u26](#), [u27](#), [u28](#),
[u29](#), [u30](#), [u34](#), [u38](#), [u43](#), [u45](#),
[u49](#), [u50](#), [u53](#), [u54](#), [u57](#), [u58](#), [u64](#)
- `\DeclareRobustCommand`
. [d198](#), [g4](#), [g11](#), [g30](#), [g57](#),
[i35](#), [i43](#), [i226](#), [i262](#), [i276](#), [i281](#),
[i296](#), [j3](#), [j13](#), [l256](#), [l257](#), [l258](#),
[l259](#), [l260](#), [l261](#), [l262](#), [l263](#), [l265](#),
[l267](#), [l269](#), [l1201](#), [m98](#), [o216](#),
[o244](#), [o245](#), [o246](#), [o250](#), [o252](#),
[o270](#), [p113](#), [s4](#), [s7](#), [s10](#), [s13](#), [s16](#),
[s19](#), [s22](#), [s25](#), [s28](#), [s34](#), [s41](#), [s89](#),
[s93](#), [t364](#), [t368](#), [t371](#), [t376](#), [t378](#),
[t380](#), [t383](#), [t389](#), [t391](#), [t393](#),
[t395](#), [t397](#), [t399](#), [t401](#), [t403](#),
[t405](#), [t407](#), [t413](#), [t415](#), [t417](#),
[t420](#), [v3](#), [v109](#), [z171](#), [z173](#), [z187](#),
[z198](#), [z248](#), [z305](#), [z322](#), [z346](#),
[B7](#), [B74](#), [B137](#), [B190](#), [B309](#),
[B334](#), [G382](#), [G390](#), [O314](#), [O321](#), [I12](#)
- `\DeclareSizeFunction` . . . [p371](#), [p444](#),
[p445](#), [p456](#), [p457](#), [p461](#), [p462](#),
[p468](#), [p469](#), [p493](#), [p494](#), [p501](#), [p502](#)
- `\DeclareSymbolFont`
. [q136](#), [r280](#), [t60](#), [t61](#), [t62](#), [t63](#)
- `\DeclareSymbolFontAlphabet`
. [r867](#), [t67](#), [t68](#), [t69](#)
- `\DeclareSymbolFontAlphabet@` [r868](#), [r871](#)
- `\DeclareTextAccent` [l64](#), [l305](#),
[l306](#), [l307](#), [l308](#), [l309](#), [l310](#), [l311](#),
[l312](#), [l313](#), [l314](#), [l315](#), [l386](#), [l387](#),
[l388](#), [l389](#), [l390](#), [l391](#), [l392](#), [l393](#),
[l394](#), [l395](#), [l396](#), [l625](#), [l630](#), [l631](#),
[l632](#), [l633](#), [l634](#), [l635](#), [l636](#), [l637](#),
[l638](#), [l639](#), [l640](#), [l719](#), [l720](#), [l721](#),
[l722](#), [l723](#), [l724](#), [l725](#), [l726](#), [l727](#),
[l728](#), [l729](#), [l730](#), [l731](#), [l732](#), [l733](#)
- `\DeclareTextAccentDefault`
. [l135](#), [l176](#), [l177](#),
[l178](#), [l179](#), [l180](#), [l181](#), [l182](#), [l183](#),
[l184](#), [l185](#), [l186](#), [l187](#), [l188](#), [l189](#),
[l229](#), [l232](#), [l1005](#), [l1006](#), [l1007](#),
[l1008](#), [l1009](#), [l1010](#), [l1011](#), [l1012](#),
[l1013](#), [l1014](#), [l1015](#), [l1016](#), [l1017](#)
- `\DeclareTextCommand` . . . [l3](#), [l58](#), [l65](#),
[l316](#), [l319](#), [l322](#), [l338](#), [l339](#), [l346](#),
[l348](#), [l350](#), [l352](#), [l358](#), [l360](#), [l362](#),
[l369](#), [l397](#), [l400](#), [l404](#), [l407](#), [l409](#),
[l412](#), [l414](#), [l614](#), [l641](#), [l643](#), [l646](#),
[l649](#), [l679](#), [l686](#), [l713](#), [l716](#), [l763](#)
- `\DeclareTextCommandDefault`
. [l57](#), [l136](#),
[l138](#), [l233](#), [l236](#), [l237](#), [l238](#), [l239](#),
[l241](#), [l245](#), [l249](#), [l250](#), [l252](#), [l253](#),
[l254](#), [l255](#), [l275](#), [l286](#), [l1071](#),
[l1073](#), [l1074](#), [l1076](#), [l1078](#), [l1080](#),
[l1082](#), [l1084](#), [l1086](#), [l1088](#), [l1090](#),
[l1092](#), [l1094](#), [l1096](#), [l1098](#), [l1100](#),
[l1102](#), [l1104](#), [l1106](#), [l1108](#), [l1110](#),
[l1112](#), [l1114](#), [l1116](#), [l1118](#), [l1120](#),
[l1122](#), [l1124](#), [l1126](#), [l1128](#), [l1130](#),
[l1132](#), [l1134](#), [l1136](#), [l1138](#), [l1140](#),
[l1142](#), [l1144](#), [l1146](#), [l1148](#), [l1150](#),
[l1152](#), [l1154](#), [l1156](#), [l1158](#), [l1160](#),
[l1162](#), [l1164](#), [l1166](#), [l1168](#), [l1170](#),
[l1172](#), [l1174](#), [l1176](#), [l1178](#), [l1180](#),
[l1182](#), [l1184](#), [l1186](#), [l1188](#), [l1191](#)
- `\DeclareTextComposite`
. [l74](#), [l376](#), [l377](#), [l471](#),
[l472](#), [l473](#), [l474](#), [l475](#), [l476](#), [l477](#),
[l478](#), [l479](#), [l480](#), [l481](#), [l482](#), [l483](#),
[l484](#), [l485](#), [l486](#), [l487](#), [l488](#), [l489](#),
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- 1490, 1491, 1492, 1493, 1494, 1495,
1496, 1497, 1498, 1499, 1500, 1501,
1502, 1503, 1504, 1505, 1506, 1507,
1508, 1509, 1510, 1511, 1512, 1513,
1514, 1515, 1516, 1517, 1518, 1519,
1520, 1521, 1522, 1523, 1524, 1525,
1526, 1527, 1528, 1529, 1530, 1531,
1532, 1533, 1534, 1535, 1536, 1537,
1538, 1539, 1540, 1541, 1542, 1543,
1544, 1545, 1546, 1547, 1548, 1549,
1550, 1551, 1552, 1553, 1554, 1555,
1556, 1557, 1558, 1559, 1560, 1561,
1562, 1563, 1564, 1565, 1566, 1567,
1568, 1569, 1570, 1571, 1572, 1573,
1574, 1575, 1576, 1577, 1578, 1579,
1580, 1581, 1693, 1694, 1695, 1696,
1697, 1698, 1699, 1700, 1701, 1702,
1703, 1704, 1705, 1706, 1707, 1708
- `\DeclareTextCompositeCommand` ...
..... 174, 1355,
1378, 1379, 1380, 1381, 1382, 1582,
1583, 1584, 1585, 1586, 1587, 1588,
1589, 1590, 1591, 1592, 1593, 1676
- `\DeclareTextFontCommand`
.. [v1](#), [v15](#), [v16](#), [v17](#), [v18](#), [v19](#),
[v20](#), [v21](#), [v22](#), [v23](#), [v24](#), [v25](#), [v123](#)
- `\DeclareTextSymbol` [l3](#),
[l325](#), [l326](#), [l327](#), [l328](#), [l329](#), [l330](#),
[l331](#), [l332](#), [l333](#), [l334](#), [l335](#), [l336](#),
[l337](#), [l340](#), [l341](#), [l342](#), [l343](#), [l344](#),
[l345](#), [l416](#), [l417](#), [l418](#), [l419](#), [l420](#),
[l421](#), [l422](#), [l423](#), [l424](#), [l425](#), [l426](#),
[l427](#), [l428](#), [l429](#), [l430](#), [l431](#), [l432](#),
[l433](#), [l434](#), [l435](#), [l436](#), [l437](#), [l438](#),
[l439](#), [l440](#), [l441](#), [l442](#), [l443](#), [l444](#),
[l445](#), [l446](#), [l447](#), [l448](#), [l449](#), [l450](#),
[l451](#), [l452](#), [l453](#), [l454](#), [l455](#), [l456](#),
[l457](#), [l458](#), [l459](#), [l460](#), [l461](#), [l462](#),
[l463](#), [l464](#), [l465](#), [l466](#), [l467](#), [l468](#),
[l469](#), [l470](#), [l601](#), [l602](#), [l603](#), [l604](#),
[l605](#), [l606](#), [l607](#), [l608](#), [l609](#), [l610](#),
[l611](#), [l612](#), [l613](#), [l623](#), [l624](#), [l652](#),
[l653](#), [l654](#), [l655](#), [l656](#), [l657](#), [l658](#),
[l659](#), [l660](#), [l661](#), [l662](#), [l663](#), [l664](#),
[l665](#), [l666](#), [l667](#), [l668](#), [l669](#), [l670](#),
[l671](#), [l672](#), [l673](#), [l674](#), [l675](#), [l734](#),
[l735](#), [l736](#), [l737](#), [l738](#), [l739](#), [l740](#),
[l741](#), [l742](#), [l743](#), [l744](#), [l745](#), [l746](#),
[l747](#), [l748](#), [l749](#), [l750](#), [l751](#), [l752](#),
[l753](#), [l754](#), [l755](#), [l756](#), [l757](#), [l758](#),
[l759](#), [l760](#), [l761](#), [l762](#), [l769](#), [l770](#),
[l771](#), [l772](#), [l773](#), [l774](#), [l775](#), [l776](#),
[l777](#), [l778](#), [l779](#), [l780](#), [l781](#), [l782](#),
[l783](#), [l784](#), [l785](#), [l786](#), [l787](#), [l788](#),
1789, 1790, 1791, 1792, 1793, 1794,
1795, 1796, 1797, 1798, 1799, 1800,
1801, 1802, 1803, 1804, 1805, 1806,
1807, 1808, 1809, 1810, 1811, 1812,
1813, 1814, 1815, 1816, 1817, 1818,
1819, 1820, 1821, 1822, 1823, 1824,
1825, 1826, 1827, 1828, 1829, 1830,
1831, 1832, 1833, 1834, 1835, 1836,
1837, 1838, 1839, 1840, 1841, 1842,
1843, 1844, 1845, 1846, 1847, 1848
- `\DeclareTextSymbolDefault`
..... [l135](#), [l190](#), [l191](#), [l192](#),
[l193](#), [l194](#), [l195](#), [l196](#), [l197](#), [l198](#),
[l199](#), [l200](#), [l201](#), [l202](#), [l203](#), [l204](#),
[l205](#), [l206](#), [l207](#), [l208](#), [l209](#), [l210](#),
[l211](#), [l212](#), [l213](#), [l214](#), [l215](#), [l216](#),
[l217](#), [l218](#), [l219](#), [l220](#), [l221](#), [l222](#),
[l223](#), [l224](#), [l225](#), [l226](#), [l227](#), [l228](#),
[l230](#), [l231](#), [l1018](#), [l1019](#), [l1020](#),
[l1021](#), [l1022](#), [l1023](#), [l1024](#), [l1025](#),
[l1026](#), [l1027](#), [l1028](#), [l1029](#), [l1030](#),
[l1031](#), [l1032](#), [l1033](#), [l1034](#), [l1035](#),
[l1036](#), [l1037](#), [l1038](#), [l1039](#), [l1040](#),
[l1041](#), [l1042](#), [l1043](#), [l1044](#), [l1045](#),
[l1046](#), [l1047](#), [l1048](#), [l1049](#), [l1050](#),
[l1051](#), [l1052](#), [l1053](#), [l1054](#), [l1055](#),
[l1056](#), [l1057](#), [l1058](#), [l1059](#), [l1060](#),
[l1061](#), [l1062](#), [l1063](#), [l1064](#), [l1065](#),
[l1066](#), [l1067](#), [l1068](#), [l1069](#), [l1070](#)
- `\default@ds`
... [L139](#), [L150](#), [L186](#), [L365](#), [L367](#)
- `\default@family`
o94, o126, o358, o361, o384, o419
- `\default@M` o101, o141, o144, [o148](#)
- `\default@mextra` [q10](#), [q89](#)
- `\default@series`
o94, o127, o359, o362, o381, o416
- `\default@shape`
o95, o128, o360, o363, o379, o414
- `\default@T` o135, o138, [o148](#), o237
- `\defaultthyphenchar` [b330](#)
- `\defaultscritratio` o479, [o486](#)
- `\defaultscriptscritratio` o480, [o486](#)
- `\defaultskewchar` [b331](#)
- `\define@mathalphabet` [q18](#), [q131](#)
- `\define@mathgroup` [q19](#), [q135](#)
- `\define@newfont` o289, [o298](#)
- `\deg` [z34](#)
- `\delcode` [r791](#)
- `\delimiter` [r722](#), [r787](#)
- `\delimiterfactor` [b332](#)
- `\delimitershortfall` [b342](#)
- `\Delta` [t217](#)
- `\delta` [t190](#)
- File Key:** a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- `\depth` B31, B34
`\det` z30
`\DH` 1418, O328
`\dh` 1428, O328
`\Diamond` s107
`\diamond` t294
`\diamondsuit` t256
`\dim` z28
`\dimen@` [b41](#), b399, b400, b436, b437, b439, b441, g28, g29, i241, i246, l353, l354, l356, l357, l677, l678, l996, l998, o179, o181, o187, o200, o203, o207, o478, o479, o480, o484, p405, p406, p407, p408, p412, z72, z73, z129, z130, z131, z132, B360, B363, C149, C150, K482, K484, K505, K507
`\dimen@i` [b41](#)
`\dimen@ii` [b41](#), o183, o188
`\dimendef` [b42](#), [b43](#), [b44](#), [b52](#), [b84](#), N216
`\dimenzero` N216
`\directlua` a9, a12, a17, a20, a25, b65, b98, b238, d21, N2, N12, N28, N207, N221, N248
`\disable_callback` [476](#), [N698](#)
`\discretionary` d10, z148
`\displ@y` z134, z138, z139
`\displaylines` [z133](#)
`\displaymath` z240
`displaymath (environment)` [z238](#)
`\displaystyle` ... t440, t443, t446, t448, z62, z140, z264, z267, z304, z329, z341, z369, z393, z396
`\displaywidowpenalty` b322
`\displaywidth` .. z140, z263, z316, z372
`\div` t297
`\DJ` 1419, O328
`\dj` 1429, O328
`\do` a69, a70, a121, b13, b14, d46, f3, f7, f16, f26, k56, k59, k99, k151, k205, k211, v73, y113, y134, y145, y151, B51, C205, C230, D31, D36, D82, D185, D187, D207, D210, D249, D293, D406, G65, G134, L78, L157, L171, L183, L188, L203, L408, L465, L524, I16, I41
`\do@noligs` [y146](#), [y151](#)
`\do@subst@correction` . [o49](#), p436, p491
`\DocInput` p8, t5, u5, M4
`\document` [81](#), [k11](#), I40
`\document@select@group` ... [r137](#), [r236](#)
File Key: a=lt`dirchk`.dtx, b=lt`plain`.dtx, c=lt`vers`.dtx, d=lt`defs`.dtx, e=lt`alloc`.dtx, f=lt`cntrl`.dtx, g=lt`error`.dtx, h=lt`par`.dtx, i=lt`space`.dtx, j=lt`logos`.dtx, k=lt`files`.dtx, l=lt`outenc`.dtx, m=lt`counts`.dtx, n=lt`length`.dtx, o=lt`fssbas`.dtx, p=lt`fssstrc`.dtx, q=lt`fssc`.dtx, r=lt`fssdcl`.dtx, s=lt`fssini`.dtx, t=lt`fontdef`.dtx, u=lt`preload`.dtx, v=lt`fntcmd`.dtx, w=lt`pageno`.dtx, x=lt`xref`.dtx, y=lt`miscen`.dtx, z=lt`math`.dtx, A=lt`lists`.dtx, B=lt`boxes`.dtx, C=lt`tab`.dtx, D=lt`pictur`.dtx, E=lt`thm`.dtx, F=lt`ssect`.dtx, G=lt`float`.dtx, H=lt`tidxglo`.dtx, I=lt`bibl`.dtx, J=lt`page`.dtx, K=lt`output`.dtx, L=lt`class`.dtx, M=lt`hyphen`.dtx, N=lt`luatex`.dtx, O=lt`final`.dtx
`\documentclass`
..... p2, t2, u2, [L207](#), [L214](#), [L241](#), [L244](#), [L333](#), [L428](#), M2, N14
`\documentstyle` [L212](#), [L428](#)
`\dorestore@version` [r114](#), [r119](#)
`\dospecials`
. a69, a121, b13, y113, y134, [L465](#)
`\dot` t433
`\doteq` t377
`\dotfill` [b443](#)
`\dots` l269, l271
`\doublehyphendemerits` b325
`\doublerulesep` C270, [C297](#), C321
`\Downarrow` t490
`\downarrow` t484
`\downbracefill` t445, t463
`\ds@` L152, L369
`\dt@pfalse` z135
`\dt@ptrue` z134
`\dump` O392
- ### E
- `\E` L466, L469, L496
`\e@alloc` b51, b52, b53, b55, b56, b63, b64, b66, b68, b76, b77, [b131](#), N13, N49, N81, N91, N180, N188, N196, N204, O12, O28
`\e@alloc@attribute@count`
. N69, N77, N78, N82, N84, N229
`\e@alloc@bytecode@count` ... N73, N192, N193, N197, N199, N245
`\e@alloc@ccodetable@count`
. N70, N87, N88, N92, N95, N233
`\e@alloc@chardef`
..... b60, [b95](#), b203, b204, N48, N180, N188, N196, N204, O12
`\e@alloc@luachunk@count` ... N74, N200, N201, N205, N209, N247
`\e@alloc@luafunction@count`
..... N71, N176, N177, N181, N183, N239, N241
`\e@alloc@top`
b55, b63, [b95](#), b181, b241, N47, N82, N181, N189, N197, N205, O12
`\e@alloc@whatsit@count` N72, N184, N185, N189, N191, N243
`\e@ch@ck` b135, [b145](#), N51, N55
`\e@insert@top` . b239, b241, b257, b272
`\e@mathgroup@top` b76, [b117](#), r56, r145
`\Eesphack` i112
`\egroup` [b380](#)
`\eject` [b410](#)
`\ell` t231
`\em` [s31](#), v25

- `\emergencystretch` [b305](#), [J45](#), [J51](#)
- `\emminershape` [s31](#)
- `\emph` [v25](#)
- `\empty` [b378](#)
- `\empty@sfcnt`
 [p444](#), [p445](#), [p446](#), [p460](#), [p465](#), [p499](#)
- `\emptyset` [t238](#)
- `\enc@update` [o222](#), [o224](#), [o240](#), [o243](#), [p129](#)
- `\encodingdefault`
 [l882](#), [l908](#), [r237](#), [s94](#), [t38](#)
- `\end` . [a64](#), [d8](#), [d287](#), [g204](#), [p9](#), [t6](#), [u6](#),
 [y60](#), [y97](#), [y98](#), [z350](#), [z359](#), [A112](#),
 [F15](#), [F17](#), [L474](#), [L478](#), [L484](#), [M5](#)
- `\end@dblfloat` [G205](#)
- `\end@float` . . [G189](#), [G227](#), [G243](#), [G363](#)
- `\endarray` [C144](#)
- `\endcenter` [y74](#)
- `\enddisplaymath` [z241](#)
- `\enddocument` [y8](#)
- `\endenumerate` [A240](#)
- `\endeqnarray` [z272](#), [z303](#)
- `\endequation` [z244](#)
- `\endfilecontents` [L432](#)
- `\endflushleft` [y81](#)
- `\endflushright` [y87](#)
- `\endgraf` [b375](#)
- `\EndIncludeInRelease`
 [a22](#), [a50](#), [b80](#), [b94](#), [b111](#), [b116](#),
 [b126](#), [b130](#), [b140](#), [b143](#), [b160](#),
 [b174](#), [b178](#), [b212](#), [b217](#), [b269](#),
 [b281](#), [b488](#), [b495](#), [b542](#), [b547](#),
 [c74](#), [c75](#), [d276](#), [d279](#), [i85](#), [i97](#),
 [i108](#), [i125](#), [i137](#), [i202](#), [i224](#), [i290](#),
 [i294](#), [l279](#), [l283](#), [l298](#), [l302](#), [m28](#),
 [m33](#), [m85](#), [m91](#), [m111](#), [m114](#),
 [n10](#), [n14](#), [o196](#), [o213](#), [o404](#),
 [o437](#), [q21](#), [q143](#), [r77](#), [r105](#), [r168](#),
 [r198](#), [s39](#), [s45](#), [z176](#), [z182](#), [z211](#),
 [z236](#), [z332](#), [z344](#), [z353](#), [z362](#),
 [A132](#), [A137](#), [B13](#), [B21](#), [B78](#),
 [B84](#), [B141](#), [B147](#), [B195](#), [B202](#),
 [B311](#), [B315](#), [B338](#), [B344](#), [D285](#),
 [D326](#), [G104](#), [G172](#), [G231](#), [G246](#),
 [G297](#), [G310](#), [G395](#), [G400](#), [K53](#),
 [K62](#), [K339](#), [K344](#), [K392](#), [K438](#),
 [K721](#), [K740](#), [K803](#), [K821](#), [K863](#),
 [K884](#), [K1126](#), [K1295](#), [K1377](#),
 [K1471](#), [K1593](#), [K1720](#), [K1839](#),
 [K1867](#), [K2131](#), [K2175](#), [N224](#),
 [N249](#), [O15](#), [O19](#), [O31](#), [O49](#),
 [O59](#), [O66](#), [O74](#), [O125](#), [O149](#)
- `\enditemize` [A251](#)
- `\endline` [b375](#), [z118](#)
- 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=ltfssstrc.dtx`, `q=ltfsscmp.dtx`, `r=ltfssdcl.dtx`, `s=ltfssini.dtx`,
`t=fontdef.dtx`, `u=preload.dtx`, `v=ltfntcmd.dtx`, `w=ltpageno.dtx`, `x=ltxref.dtx`,
`y=ltmiscen.dtx`, `z=ltmath.dtx`, `A=ltlists.dtx`, `B=ltboxes.dtx`, `C=lttab.dtx`,
`D=ltpictur.dtx`, `E=ltthm.dtx`, `F=ltsect.dtx`, `G=ltfloat.dtx`, `H=ltidxglo.dtx`,
`I=ltbibl.dtx`, `J=ltpage.dtx`, `K=ltoutput.dtx`, `L=ltclass.dtx`, `M=lthyphen.dtx`,
`N=ltluatex.dtx`, `O=ltfinal.dtx`
- `\endlinechar` [a87](#), [a88](#), [a89](#), [a199](#), [d24](#),
 [d26](#), [d31](#), [k179](#), [L112](#), [L113](#), [L114](#)
- `\endlist` [A98](#), [A240](#), [A251](#)
- `\endlrbox` [B108](#)
- `\endmath` [z239](#)
- `\endminipage` [B277](#)
- `\endpicture` [D17](#)
- `\endsloppypar` [J49](#)
- `\endtabbing` [C73](#)
- `\endtabular` [C144](#)
- `\endtabular*` [C144](#)
- `\endtrivlist` [y74](#), [y81](#), [y87](#),
 [y119](#), [z374](#), [A100](#), [A101](#), [C74](#), [E39](#)
- `\endverbatim` [y118](#), [y122](#)
- `\enlargethispage` [K1765](#)
- `\enlargethispage*` [K1765](#)
- `\enskip` [i306](#)
- `\enspace` [i303](#)
- `\ensuremath`
 [m87](#), [z305](#), [G385](#), [G393](#), [G403](#)
- `\enumerate` [A231](#)
- `enumerate (environment)` [A231](#)
- environments:
 `center` [y73](#)
 `displaymath` [z238](#)
 `enumerate` [A231](#)
 `eqnarray` [z250](#), [z375](#)
 `eqnarray*` [z301](#)
 `equation` [z242](#), [z363](#)
 `filecontents` [453](#)
 `flushleft` [y80](#)
 `flushright` [y86](#)
 `itemize` [A242](#)
 `lrbox` [284](#)
 `math` [z238](#)
 `minipage` [285](#)
 `sloppypar` [J48](#)
 `thebibliography` [376](#)
 `verbatim*` [y121](#)
- `\epsilon` [t191](#)
- `\eqnarray` [z255](#), [z302](#)
- `eqnarray (environment)` [z250](#), [z375](#)
- `eqnarray* (environment)` [z301](#)
- `\eqno` [z244](#)
- `\equation` [z243](#)
- `equation (environment)` [z242](#), [z363](#)
- `\equiv` [t356](#)
- `\err@rel@i` [q12](#), [q99](#), [q132](#), [q136](#)
- `\errhelp` [a212](#),
 [c30](#), [g39](#), [g66](#), [M12](#), [O223](#), [O383](#)
- `\errmessage`
 [a4](#), [a217](#), [b157](#), [b171](#), [b285](#),

- c31, g47, g72, o376, o411, p379,
 p479, q65, M16, N63, O43, O225
 \error@fontshape
 o353, o377, o412, p107, p481, r222
 \errorcontextlines ... b303, b335,
 b462, b478, b493, b506, b523, g163
 \errorstopmode b451, O391
 \escapechar d103, d146,
 d150, d158, o301, o446, p183,
 r58, r86, r147, r177, r221, N206
 \et@xmaxfam N20, N24, N30, N38
 \et@xmaxregs N28,
 N31, N32, N33, N34, N35, N36, N37
 \eta t193
 \etatcatcode N732
 \evensidemargin K80, K584
 \every@math@size o43, p189, p201
 \everycr .. b431, z135, z138, z263, z390
 \everydisplay o279, o280, o285
 \everyjob c36, c41, c46, r241,
 N210, N211, O353, O354, O356
 \everymath o278, o280, o283
 \everypar 64, k37,
 o494, y50, y116, A129, A131,
 A135, A136, A180, A197, B238,
 C70, F31, F79, F90, F110,
 F119, G187, K168, K1061, K1227
 \execute@size@function
 p316, p344, p358, p375
 \ExecuteOptions . l944, p57, p70, L201
 \exhyphenpenalty b317, b404
 \exists t249
 \exp z31
 \external@font p84,
 p87, p98, p102, p104, p345,
 p359, p421, p455, p505, p507, p509
 \extra@def q9, q84
 \extracolsep C140
 \extract@alph@from@version
 o452, o458, r151, r182
 \extract@font o312, p81
 \extract@fontinfo p312, p319
 \extract@rangefontinfo
 p329, p336, p355, p388
 \extract@sizefn p304, p326
 \extrafloats b145, b182, b255
- F**
- \f@baselineskip
 o251, o258, o365, p119,
 p136, p140, p155, p169, p180, p194
 \f@depth G295, K319
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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx
- \f@encoding
 . l128, o216, o235, o238, o239,
 o241, o260, o292, o297, o316,
 o318, o320, o325, o327, o357,
 o373, o408, p91, p261, p471, r207
 \f@family ... 1961, 1964, 1978, 1988,
 1994, 11209, o244, o254, o293,
 o297, o316, o318, o320, o325,
 o327, o361, o384, o419, p91, r207
 \f@linespread ... o254, p118, p137,
 p138, p141, p149, p152, p163, p166
 \f@series j14, o244, o255,
 o294, o297, o362, o381, o416, s81
 \f@shape o244,
 o256, o295, o297, o363, o379, o414
 \f@size l130, 1996, o53, o251,
 o257, o296, o364, o401, o435,
 o477, o478, o481, o482, p119,
 p121, p134, p154, p169, p172,
 p175, p180, p187, p194, p206,
 p209, p215, p221, p238, p239,
 p242, p247, p313, p320, p339,
 p341, p356, p407, p409, p411,
 p427, p428, p433, p447, p459,
 p464, p476, p484, p489, p497, p511
 \f@user@size .. p427, p432, p476, p489
 \fam b91, o16, N20, N24, N38
 \familydefault r238, s95, t38
 \fbox 284, B126, B139, B146
 \fboxrule B124, B159, B162,
 B168, B170, B177, B178, O78
 \fboxsep B124, B130,
 B158, B163, B173, B175, O77
 \filbreak b408
 \filecontents
 ... L437, L438, L439, L518, L524
 \filecontents L432
 filecontents (environment) 453
 \filename@area a241, a247,
 a254, a260, a267, a273, a280,
 k168, k190, k193, k207, k219, k221
 \filename@base
 a289, k168, k190, k193, k214, k219
 \filename@dot a287, a290
 \filename@ext a285, a287,
 k169, k186, k187, k190, k193, k215
 \filename@parse 1,
 6, a105, a237, k166, k185, k212
 \filename@path .. a242, a243, a248,
 a255, a256, a261, a268, a269, a274
 \filename@simple
 a245, a258, a271, a281, a283
 \fill i300
 \finalhyphendemerits b326

`\finph@nt` z87, z89, z90 K2008, K2016, K2018, K2025,
`\finsm@sh` z103, z105, z106 K2030, K2035, K2037, K2043,
`\firstmark` J37, K618, K2106 K2045, K2052, K2079, K2081,
`\fix@penalty` v84 K2093, K2118, K2122, K2127,
`\fixed@sfcnt` p501, p502, p503 K2139, K2156, K2161, K2169
`\fl@trace` K214, K241, K297, K325,
 K332, K353, K400, K446, K499,
 K514, K515, K516, K517, K528,
 K529, K530, K531, K532, K542,
 K691, K710, K729, K747, K749,
 K888, K892, K904, K905, K906,
 K907, K913, K916, K924, K928,
 K939, K944, K949, K950, K951,
 K952, K959, K962, K970, K981,
 K987, K992, K997, K1003,
 K1004, K1009, K1014, K1015,
 K1016, K1024, K1028, K1033,
 K1037, K1042, K1053, K1054,
 K1056, K1074, K1083, K1089,
 K1098, K1101, K1107, K1117,
 K1121, K1131, K1137, K1143,
 K1149, K1156, K1158, K1164,
 K1169, K1171, K1173, K1181,
 K1186, K1192, K1197, K1203,
 K1217, K1218, K1221, K1242,
 K1251, K1257, K1266, K1269,
 K1276, K1286, K1290, K1302,
 K1308, K1313, K1318, K1322,
 K1326, K1327, K1334, K1339,
 K1343, K1350, K1359, K1363,
 K1367, K1368, K1372, K1373,
 K1383, K1389, K1395, K1401,
 K1405, K1411, K1413, K1421,
 K1426, K1431, K1439, K1448,
 K1453, K1458, K1460, K1465,
 K1467, K1478, K1484, K1494,
 K1500, K1504, K1505, K1510,
 K1511, K1517, K1520, K1521,
 K1522, K1529, K1530, K1531,
 K1539, K1544, K1556, K1557,
 K1564, K1567, K1575, K1579,
 K1583, K1584, K1588, K1589,
 K1599, K1605, K1615, K1621,
 K1625, K1626, K1632, K1633,
 K1640, K1643, K1644, K1645,
 K1653, K1654, K1655, K1664,
 K1669, K1682, K1684, K1691,
 K1694, K1703, K1707, K1711,
 K1712, K1716, K1717, K1769,
 K1774, K1780, K1790, K1797,
 K1807, K1903, K1916, K1917,
 K1921, K1924, K1926, K1929,
 K1932, K1934, K1975, K1982,
 K1987, K1993, K1998, K2002,
`\fl@tracemessage` K1807
`\fl@traceval` K1807
`\flat` t252
`\float@count` .. b51, b52, b53, b62,
 b181, b198, b203, b205, b206, b215
`\floatingpenalty` G418
`\floatpagefraction` G282, K2187
`\floatsep` K636,
 K654, K661, K2014, K2064, K2192
`\flushbottom` J41
`\flushleft` y80
`flushleft` (environment) y80
`\flushright` y86
`flushright` (environment) y86
`\fmtname` c1, c37,
 c39, c42, c44, c47, c49, L250, L254
`\fmtversion` .. c1, c18, c37, c39, c42,
 c44, c47, c49, c62, g2, o1, C1,
 D1, K4, L267, L270, O337, O363
`\fmtversion@topatch` O335,
 O337, O349, O350, O362, O370
`\fnsymbol` 132, m50
`\font` b436, b441,
 l246, l247, l248, l363, l370, l680,
 l687, o46, o52, o54, p84, s35,
 s42, s68, s80, u8, u9, u10, v68, y115
`\font@info` p99, p319, p388, p393
`\font@name` l129,
 l132, o51, o159, o161, o288,
 o303, o400, o434, p84, p88,
 p90, p105, p120, p123, p126,
 p284, p285, p286, p287, p288, p293
`\font@submax` p395, p424,
 p425, y22, y24, O215, O217, O226
`\fontdimen` b436, b441, l246,
 l247, l248, l363, l370, l680, l687,
 s35, s42, s80, v68, D38, D40, D364
`\fontencoding` 1908, o216, o247, r237, t14
`\fontfamily` 1980, o244, r238, s6, s9, s12
`\fontname` o54
`\fontseries` o244, r239, s15, s18
`\fontshape` l373,
 l690, o244, r240, s21, s24, s27, s30
`\fontsize` . j6, l251, l277, l291, 1998,
 o44, o252, s74, G385, G393, G403
`\fontsubfuzz` p395, p429, y22
`\footins` G370, G414,
 K288, K289, K290, K291, K349,
 K396, K456, K464, K468, K491

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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- `\footnote` [G405](#)
`\footnotemark` [F9](#), [G427](#)
`\footnoterule` [B283](#), [G374](#), [K467](#)
`\footnotesep` . [B302](#), [G404](#), [G417](#), [G425](#)
`\footnotesize` [B295](#), [G415](#)
`\footnotetext` [F11](#), [G444](#)
`\footskip` [K84](#), [K608](#)
`\forall` [t248](#)
`\fps@dbl` [G34](#)
`\frac` [z247](#)
`\frame` [B110](#), [B185](#)
`\framebox` [284](#), [B133](#)
`\frenchspacing` . [b361](#), [k40](#), [y118](#), [y144](#)
`\frown` [t359](#)
`\frozen@everydisplay` [o278](#), [o284](#)
`\frozen@everymath` [o278](#), [o282](#)
`\fussy` [J50](#)
`\futurelet` [d293](#),
[d307](#), [i266](#), [i274](#), [v66](#), [z153](#), [C318](#)
- G**
- `\g@addto@macro` [L376](#), [L382](#), [L386](#), [L387](#)
`\G@refundefinedfalse` [x5](#)
`\G@refundefinedtrue` . [x3](#), [x12](#), [I21](#), [I44](#)
`\Gamma` [t216](#)
`\gamma` [t189](#)
`\gcd` [z33](#)
`\ge` [t330](#)
`\gen@sfcnt` [p456](#), [p457](#), [p458](#)
`\genb@sfcnt` [p461](#), [p462](#), [p463](#)
`\genb@x` [p464](#), [p466](#)
`\genb@y` [p466](#)
`\GenericError` [g18](#), [g85](#), [g111](#), [g137](#), [p62](#)
`\GenericInfo` [c63](#), [c66](#), [c70](#), [g4](#),
[g104](#), [g130](#), [g155](#), [p31](#), [p34](#), [p39](#), [p75](#)
`\GenericWarning` [g11](#),
[g94](#), [g120](#), [g146](#), [p42](#), [p47](#), [p50](#), [p78](#)
`\geq` [t329](#), [t330](#)
`\get@cdp` [r356](#), [r364](#), [r397](#)
`\get@external@font` ... [p83](#), [p96](#), [p490](#)
`\getanddefine@fonts` .. [o447](#), [o465](#),
[p274](#), [r59](#), [r87](#), [r132](#), [r148](#), [r178](#),
[r263](#), [r327](#), [r361](#), [r363](#), [r380](#),
[r503](#), [r504](#), [r536](#), [r537](#), [r887](#), [r888](#)
`\GetFileInfo` [t3](#)
`\getlinechar` [D108](#)
`\gets` [t348](#)
`\gg` [t343](#)
`\glb@currsz` [k35](#),
[o275](#), [p171](#), [p206](#), [p210](#), [p216](#), [p239](#)
`\glb@settings` . [o276](#), [p171](#), [p218](#), [p249](#)
`\globaldefs`
... [o448](#), [p185](#), [r60](#), [r89](#), [r149](#), [r180](#)
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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- `\glossary` [374](#),
[F146](#), [H23](#), [H35](#), [J20](#), [J28](#), [K592](#)
`\glossaryentry` [H32](#)
`\goodbreak` [b408](#)
`\grave` [t425](#)
`\group@elt` [r35](#),
[r261](#), [r298](#), [r299](#), [r320](#), [r324](#), [r919](#)
`\group@list`
... [r265](#), [r305](#), [r318](#), [r323](#), [r324](#),
[r353](#), [r575](#), [r618](#), [r699](#), [r702](#),
[r753](#), [r756](#), [r804](#), [r807](#), [r874](#), [r925](#)
`\guillemotleft` [l430](#), [l657](#)
`\guillemotright` [l431](#), [l658](#)
`\guilsinglleft` [l432](#)
`\guilsinglright` [l433](#)
- H**
- `\H` [g24](#), [l180](#), [l312](#),
[l391](#), [l485](#), [l493](#), [l512](#), [l520](#), [l637](#)
`\h@false` [z77](#)
`\h@true` [z78](#), [z79](#)
`\halign` ... [b431](#), [z96](#), [z140](#), [z263](#), [z390](#)
`\hangindent` [F122](#)
`\hat` [t431](#)
`\hb@xt@` [b446](#), [d16](#), [l349](#),
[z140](#), [z268](#), [z314](#), [z329](#), [z341](#),
[z368](#), [z398](#), [B43](#), [B58](#), [B157](#),
[B368](#), [B372](#), [B373](#), [C37](#), [D13](#),
[D23](#), [D32](#), [D122](#), [D156](#), [D159](#),
[D162](#), [D164](#), [D166](#), [D237](#), [D278](#),
[D321](#), [D416](#), [F163](#), [F166](#), [K601](#),
[K611](#), [K1753](#), [K2112](#), [K2113](#),
[K2117](#), [K2144](#), [K2145](#), [K2151](#)
`\hbadness` [b313](#), [o497](#)
`\hbar` [t228](#)
`\headheight` [K82](#), [K597](#)
`\headsep` [K83](#), [K606](#)
`\heartsuit` [t257](#)
`\height` [B30](#), [B33](#)
`\hexnumber@` [r591](#),
[r599](#), [r614](#), [r634](#), [r642](#), [r650](#),
[r659](#), [r662](#), [r671](#), [r672](#), [r711](#),
[r719](#), [r765](#), [r773](#), [r787](#), [r788](#),
[r791](#), [r817](#), [r825](#), [r830](#), [r832](#), [s85](#)
`\hfuzz` [b336](#), [J46](#), [J47](#), [J53](#), [J54](#)
`\hgl@` [b401](#), [b402](#)
`\hglue` [b398](#)
`\hideoutput` [b496](#)
`\hideskip` [b288](#), [b422](#)
`\hidewidth` [b422](#), [l276](#), [l278](#),
[l289](#), [l293](#), [l317](#), [l318](#), [l321](#), [l324](#),
[l398](#), [l399](#), [l403](#), [l406](#), [l408](#), [l411](#),
[l644](#), [l645](#), [l648](#), [l651](#), [l715](#), [l718](#)
`\hline` [C317](#), [C320](#)

`\hmode@bgroup` 167, 173, 1276,
 1287, 1317, 1323, 1351, 1362, 1369,
 1398, 1405, 1408, 1410, 1614, 1644,
 1650, 1679, 1686, 1714, 1717, 1763, v7
`\hmode@start@before@group`
 168, 1111, 1113, 1119, 1134
`\holdinginserts` b304
`\hom` z29
`\hookleftarrow` t388
`\hookrightarrow` t386
`\hphantom` z75
`\hrule` b399, b443,
 i242, i250, 1240, 1243, t246,
 t522, B116, B121, B168, B178,
 C318, C335, D280, D323, G375
`\hrulefill` b443
`\hspace` i296
`\hyphenation` 1155
`\hyphenchar` y115
`\hyphenpenalty` b316

I

`\I` b367, L492, L510, O186, O302
`\i` 1197,
 1329, 1376, 1377, 1378, 1379, 1380,
 1381, 1434, 1471, 1472, 1564, 1566,
 1568, 1570, 1659, O191, O307, O315
`\ialign` b431, b433,
 t243, t367, t438, t441, t444,
 t447, z109, z111, z119, C164, D51
`\if@afterindent` F107, F114
`\if@compatibility` L2, L209
`\if@endpe` y62, A138
`\if@eqnsw` z250, z299
`\if@fcolmade`
 . . . K102, K238, K368, K377,
 K415, K425, K689, K709, K727,
 K756, K836, K2078, K2126, K2166
`\if@filesw` k7,
 k30, k92, k104, k111, k120, y14,
 y28, F136, I4, I8, I19, I28, I36, I43
`\if@firstamp` C212
`\if@firstcolumn` K102, K220, K253,
 K370, K418, K1725, K2090, K2135
`\if@ignore` y4, y63
`\if@inlabel`
 A28, A65, A102, A160, A183, K164
`\if@insert` K102, K967,
 K1079, K1113, K1247, K1282,
 K1356, K1445, K1572, K1700
`\if@minipage` i155, i172,
 i207, y101, A149, B246, C68, G20
`\if@mparswitch` K102, K1727
`\if@multiplelabels` x31
`\if@negarg` D55, D77, D91, D130
`\if@newlist` y119, A29, A33,
 A69, A78, A106, A166, K568, K615
`\if@nbrlist` A33, A201
`\if@no@font@opt` q16, q110, q129
`\if@nobreak` i58, i174, i209,
 k67, k79, A167, A192, B232,
 F30, F111, G180, G353, J25,
 J33, K168, K309, K1058, K1224
`\if@noitemarg` A32, A199
`\if@noparitem` A30, A157
`\if@noparlist` A31, A114
`\if@noskipsec` A58,
 B233, F21, F23, F80, G354, K158
`\if@ovb` D212, D265, D270, D309, D314
`\if@ovhline` D244, D280, D290
`\if@ovl` D212, D263, D282, D305, D324
`\if@ovr` D212, D262, D279, D304, D322
`\if@ovt` D212, D264, D275, D308, D318
`\if@ovvline` D244, D273, D289
`\if@partsw` k7, k96
`\if@pboxsw` B229, B304
`\if@reversemargin` K108, K1730
`\if@reversemarginpar` K102
`\if@rjfield` C19, C33
`\if@specialpage` K102, K577
`\if@tempwa` a73, a74, a75, b251,
 e9, k102, o64, r286, r341, r405,
 r486, r918, y30, y107, K900,
 K936, K1536, K1661, L455, I52
`\if@test` K12, K13,
 K797, K816, K856, K878, K942,
 K1026, K1035, K1184, K1195,
 K1337, K1424, K1542, K1667
`\if@twocolumn`
 k20, G32, G210, G235, K102,
 K146, K241, K252, K369, K417,
 K441, K691, K747, K1724, K2080
`\if@twoside` K102, K145, K580
`\ifdt@p` z133, z135
`\iff` t408
`\ifFileExists` 81,
 455, a173, k134, k161, k172, O331
`\ifG@refundefined` x3, x4, x5
`\ifh@` z76, z93
`\ifin@` l898, l901, q50, q52, r1,
 r22, r250, r352, r354, r415, r428,
 r498, r500, r528, r576, r588,
 r619, r631, r700, r703, r724,
 r754, r757, r802, r805, r808,
 r875, r877, r906, L82, L162, L174
`\ifinner` z174,
 z181, z200, z226, G57, G126, G319
`\ifmath@fonts` o169, p176

File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx

- `\ifmaybe@ic` [v65](#), [v74](#)
`\ifnot@nil` [p297](#), [p314](#), [p335](#)
`\ifodd` [r849](#),
[D171](#), [D191](#), [G68](#), [G137](#), [K21](#),
[K145](#), [K581](#), [K892](#), [K895](#), [K928](#),
[K931](#), [K1042](#), [K1045](#), [K1204](#),
[K1207](#), [K1484](#), [K1487](#), [K1605](#),
[K1608](#), [K1728](#), [K1949](#), [K1957](#)
`\iftc@forced` [l936](#), [l946](#), [l1215](#)
`\ifv@` [z75](#), [z92](#)
`\ifvbox` [K293](#), [K350](#), [K397](#), [K476](#), [K492](#)
`\ignorespaces` [i24](#),
[i81](#), [i94](#), [i105](#), [i121](#), [i134](#), [i312](#),
[k60](#), [o249](#), [y63](#), [y71](#), [y72](#), [z208](#),
[z234](#), [A55](#), [A217](#), [B107](#), [B302](#),
[C57](#), [C58](#), [C59](#), [C72](#), [C81](#), [C94](#),
[C98](#), [C105](#), [C112](#), [C114](#), [C123](#),
[C198](#), [C260](#), [C262](#), [C264](#), [C291](#),
[D16](#), [D24](#), [D35](#), [D53](#), [D54](#), [E30](#),
[E32](#), [F93](#), [G17](#), [G24](#), [G425](#), [I7](#), [I9](#)
`\ignorespacesafterend` [y7](#)
`\IJ` [l200](#), [l360](#), [l437](#)
`\ij` [l199](#), [l358](#), [l436](#)
`\Im` [t234](#)
`\imath` [t229](#)
`\in` [t340](#), [t369](#)
`\in@` [l896](#), [l899](#), [q49](#), [q51](#), [r1](#),
[r21](#), [r249](#), [r351](#), [r353](#), [r411](#), [r424](#),
[r497](#), [r499](#), [r526](#), [r574](#), [r585](#),
[r617](#), [r629](#), [r698](#), [r701](#), [r721](#),
[r752](#), [r755](#), [r799](#), [r803](#), [r806](#),
[r873](#), [r876](#), [r904](#), [L81](#), [L159](#), [L173](#)
`\in@@` [r5](#), [r6](#), [r7](#), [r9](#)
`\in@false` [r10](#)
`\in@true` [r12](#)
`\in_callback` [476](#), [N682](#)
`\include` [81](#), [k86](#)
`\IncludeInRelease`
[a18](#), [a23](#), [b49](#), [b81](#), [b96](#),
[b112](#), [b118](#), [b127](#), [b132](#), [b141](#),
[b147](#), [b161](#), [b175](#), [b179](#), [b213](#),
[b226](#), [b270](#), [b454](#), [b489](#), [b496](#),
[b543](#), [c53](#), [d249](#), [d277](#), [i70](#), [i86](#),
[i98](#), [i111](#), [i126](#), [i167](#), [i203](#), [i285](#),
[i291](#), [l273](#), [l281](#), [l284](#), [l300](#), [m24](#),
[m30](#), [m70](#), [m86](#), [m94](#), [m112](#),
[n5](#), [n11](#), [o175](#), [o197](#), [o369](#), [o405](#),
[q2](#), [q22](#), [r49](#), [r78](#), [r138](#), [r169](#),
[s32](#), [s40](#), [z169](#), [z177](#), [z185](#),
[z212](#), [z321](#), [z333](#), [z345](#), [z354](#),
[A125](#), [A133](#), [B4](#), [B14](#), [B71](#),
[B79](#), [B134](#), [B142](#), [B187](#), [B196](#),
[B306](#), [B312](#), [B331](#), [B339](#), [D240](#),
[D286](#), [G35](#), [G105](#), [G206](#), [G232](#),
[G284](#), [G298](#), [G387](#), [G396](#),
[K24](#), [K54](#), [K319](#), [K340](#), [K345](#),
[K393](#), [K704](#), [K722](#), [K783](#), [K804](#),
[K840](#), [K864](#), [K976](#), [K1127](#),
[K1296](#), [K1378](#), [K1472](#), [K1594](#),
[K1813](#), [K1840](#), [K2086](#), [K2132](#),
[N3](#), [N227](#), [O8](#), [O16](#), [O23](#),
[O32](#), [O51](#), [O60](#), [O67](#), [O93](#), [O126](#)
`\includeonly` [81](#), [k82](#)
`\indent` [A161](#), [C70](#)
`\index` [374](#), [F146](#), [H6](#), [H18](#), [J20](#), [J28](#), [K591](#)
`\indexentry` [H15](#)
`\inf` [z25](#)
`\infty` [t236](#)
`\init@restore@glb@settings`
..... [p219](#), [p222](#), [p224](#)
`\init@restore@version`
..... [r62](#), [r91](#), [r108](#), [r123](#), [r124](#)
`\initcatcodetable` [N93](#)
`\input` [81](#), [456](#), [a63](#), [a169](#), [a172](#), [a229](#),
[d7](#), [k163](#), [l1194](#), [p16](#), [q106](#), [s145](#),
[s156](#), [s166](#), [t10](#), [t11](#), [t12](#), [t13](#),
[t20](#), [t21](#), [t25](#), [t26](#), [t55](#), [t56](#),
[t57](#), [t58](#), [t540](#), [t541](#), [t542](#), [L213](#),
[N16](#), [O91](#), [O105](#), [O130](#), [O212](#), [O336](#)
`\input@path` [1](#), [6](#), [a104](#), [a126](#),
[a128](#), [a134](#), [a136](#), [a142](#), [a144](#),
[a149](#), [a151](#), [a161](#), [a228](#), [k137](#), [k151](#)
`\InputIfFileExists`
... [81](#), [455](#), [k160](#), [k165](#), [k173](#),
[k189](#), [l886](#), [l1277](#), [o325](#), [s119](#),
[s137](#), [s148](#), [s158](#), [L339](#), [M8](#), [O206](#)
`\inputlineno` [a298](#), [b295](#), [b296](#), [b297](#),
[g165](#), [g168](#), [s118](#), [O171](#), [O182](#),
[O190](#), [O271](#), [O287](#), [O298](#), [O306](#)
`\insecunt`
. [b37](#), [b51](#), [b52](#), [b53](#), [b62](#), [b83](#),
[b84](#), [b85](#), [b87](#), [b229](#), [b230](#), [b231](#),
[b232](#), [b233](#), [b234](#), [b245](#), [b246](#),
[b247](#), [b248](#), [b252](#), [b253](#), [b273](#),
[b274](#), [b275](#), [b276](#), [b277](#), [b278](#), [K61](#)
`\insert` ... [b236](#), [b260](#), [b262](#), [b265](#),
[b280](#), [G414](#), [K491](#), [K492](#), [K1793](#)
`\install@mathalphabet`
. [o442](#), [o459](#), [o466](#), [r269](#), [r272](#),
[r358](#), [r359](#), [r456](#), [r508](#), [r511](#),
[r518](#), [r533](#), [r534](#), [r541](#), [r889](#), [r891](#)
`\int` [t266](#)
`\interdisplaylinepenalty`
..... [i29](#), [z55](#), [z137](#), [z285](#)
`\interfootlinepenalty` [b357](#)
`\interfootnotelinepenalty`
..... [b357](#), [i34](#), [G416](#)
- File Key:** a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- `\interlinepenalty` [i27](#), [y108](#), [y111](#),
[F50](#), [F101](#), [F154](#), [G416](#), [K312](#),
[K1063](#), [K1067](#), [K1229](#), [K1233](#)
`\intertextsep` . [K1046](#), [K1050](#), [K1065](#),
[K1068](#), [K1075](#), [K1208](#), [K1214](#),
[K1231](#), [K1234](#), [K1243](#), [K2192](#)
`\intop` [t265](#), [t266](#)
`\iota` [t195](#)
`\isorange` [p330](#), [p331](#)
`\ishortstack` [D42](#)
`\itdefault` [s30](#), [t34](#)
`\item` [g234](#), [y73](#), [y80](#),
[y86](#), [y100](#), [z328](#), [z340](#), [z367](#),
[A141](#), [A219](#), [C67](#), [E36](#), [E38](#), [I4](#), [I8](#)
`\itemindent` . [A9](#), [A42](#), [A95](#), [A187](#), [A208](#)
`\itemize` [A242](#)
`itemize (environment)` [A242](#)
`\itemsep` [A1](#), [A176](#)
`\iterate` [a76](#), [a77](#), [b387](#)
`\itshape` [l371](#), [l688](#), [s28](#),
[s29](#), [s36](#), [s43](#), [v21](#), [E36](#), [E38](#), [G379](#)
- J**
- `\J` [O188](#), [O304](#)
`\j` [l198](#), [l330](#), [l435](#), [l660](#), [O315](#)
`\jmath` [t230](#)
`\Join` [s105](#)
`\joinrel` [t379](#), [t386](#), [t388](#), [t390](#), [t392](#),
[t394](#), [t396](#), [t398](#), [t400](#), [t404](#), [t406](#)
`\jot` [z53](#), [z134](#), [z292](#)
- K**
- `\k` [l407](#),
[l474](#), [l479](#), [l501](#), [l506](#), [l582](#), [l583](#),
[l641](#), [l642](#), [l693](#), [l695](#), [l700](#), [l702](#)
`\kappa` [t196](#)
`\ker` [z27](#)
`\kernel@ifnextchar` [c54](#),
[d58](#), [d77](#), [d127](#), [d294](#), [d311](#), [L119](#)
`\kill` [C59](#)
- L**
- `\L` [l192](#), [l348](#), [l420](#), [l655](#), [L489](#), [L509](#), [O328](#)
`\l` [l201](#), [l350](#), [l438](#), [l661](#), [O328](#)
`\l@ngrel@x` [d51](#), [d52](#), [d53](#), [d97](#), [d144](#)
`\label` [x32](#), [F146](#), [J20](#), [J28](#), [K590](#)
`\labelsep` [A9](#), [A210](#), [A216](#), [E36](#), [E38](#)
`\labelwidth` [A9](#), [A93](#), [A209](#), [A211](#), [A214](#)
`\Lambda` [t219](#)
`\lambda` [t197](#)
`\land` [t283](#)
`\langle` [t498](#)
`\language` [b35](#), [b77](#), [b92](#), [b298](#), [b299](#), [M10](#)
`\last@fontshape` [o375](#), [o393](#), [o410](#), [o427](#)
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=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- `\lastbox` [z123](#), [z124](#), [A130](#),
[A136](#), [A185](#), [F82](#), [F115](#), [K279](#)
`\LastDeclaredEncoding` [o102](#), [o105](#)
`\lastpenalty` [v95](#), [v98](#)
`\lastskip` [b411](#),
[b412](#), [b414](#), [b416](#), [i19](#), [i66](#), [i78](#),
[i140](#), [i141](#), [i145](#), [i147](#), [i148](#), [i156](#),
[i176](#), [i179](#), [i211](#), [i214](#), [i215](#), [v85](#),
[v88](#), [A115](#), [A116](#), [A150](#), [A151](#), [D36](#)
`\LaTeX` [j3](#), [j15](#), [L458](#)
`\LaTeXe` [j13](#)
`\latexreleaseversion` [c5](#)
`\lbrace` [l257](#), [t502](#)
`\lbrack` [b371](#)
`\lccode` [g19](#),
[g20](#), [g21](#), [g22](#), [g23](#), [g24](#), [l104](#),
[y139](#), [y149](#), [O151](#), [O168](#), [O178](#),
[O187](#), [O189](#), [O191](#), [O193](#), [O196](#),
[O197](#), [O198](#), [O199](#), [O284](#), [O294](#),
[O303](#), [O305](#), [O307](#), [O309](#), [O312](#)
`\lceil` [t506](#)
`\ldotp` [t409](#), [t412](#), [t523](#)
`\ldots` [l271](#), [t413](#)
`\le` [t328](#)
`\leaders` [b443](#), [t246](#),
[t464](#), [t465](#), [t467](#), [t468](#), [C335](#),
[D273](#), [D280](#), [D317](#), [D323](#), [F159](#)
`\leadsto` [s108](#)
`\leavevmode` [b402](#), [b429](#),
[b432](#), [b443](#), [b445](#), [i263](#), [i277](#),
[l73](#), [l134](#), [l238](#), [l240](#), [l320](#), [l349](#),
[l353](#), [l356](#), [l401](#), [l647](#), [l677](#), [l992](#),
[v106](#), [y108](#), [y119](#), [y132](#), [y150](#),
[z328](#), [z340](#), [z367](#), [A58](#), [A103](#),
[B8](#), [B17](#), [B23](#), [B109](#), [B111](#),
[B127](#), [B154](#), [B214](#), [B262](#), [B318](#),
[B335](#), [B342](#), [C151](#), [D44](#), [D166](#),
[F23](#), [F155](#), [G439](#), [K160](#), [K165](#), [I14](#)
`\left` [t524](#),
[t525](#), [t526](#), [t527](#), [z108](#), [z114](#), [z125](#)
`\Leftarrow` [t324](#), [t400](#), [t406](#)
`\leftarrow`
. [t347](#), [t348](#), [t388](#), [t398](#), [t404](#), [t456](#)
`\leftarrowfill` [t442](#), [t456](#)
`\lefteqn` [z304](#)
`\leftharpoondown` [t361](#), [t375](#)
`\leftharpoonup` [t360](#)
`\lefthyphenmin` [b301](#), [M11](#)
`\leftline` [B368](#)
`\leftmargin`
. [A9](#), [A52](#), [A53](#), [A94](#), [A146](#), [A148](#)
`\leftmargini` [z320](#), [A17](#)
`\leftmarginii` [A17](#)
`\leftmarginiii` [A17](#)

- `\leftmarginiv` [A17](#)
`\leftmarginv` [A17](#)
`\leftmarginvi` [A17](#)
`\leftmark` [J34](#)
`\Leftrightarrow` [t323](#)
`\leftrightharpoonrightarrow` [t346](#)
`\leftskip` [b424, y77, y84, y90, y102, A74, B241, F152, F157](#)
`\leq` [t327, t328](#)
`\lfloor` [t510](#)
`\lg` [z4](#)
`\lgroup` [t512](#)
`\lhd` [s111](#)
`\lhook` [t385, t386](#)
`\lim` [z6](#)
`\liminf` [z8](#)
`\limits` [t446, t450, z107, z246](#)
`\limsup` [z7](#)
`\line` [g223, D56, D235](#)
`\linebreak` [66, i13](#)
`\linepenalty` [b315](#)
`\lineskip`
 . [b365, b397, b432, t366, z130, B242, C60, C171, D46, D167, K593](#)
`\lineskiplimit` .. [b366, b397, b434, b435, t366, t418, z132, z136, K593](#)
`\linespread` [o250](#)
`\linethickness` [D41](#)
`\linewidth` [k24, z193, z219, z329, z341, z368, z372, z390, A15, A51, A52, A54, B239, C36, G270, K153, K179](#)
`\list` [A34, A236, A247](#)
`\listfiles` [456, k201](#)
`\listparindent` [A9, A41, A50](#)
`\ll` [t344](#)
`\llap` [A238, A249, B372, B373](#)
`\lmoustache` [t469](#)
`\ln` [z5](#)
`\lnot` [t251](#)
`\LoadClass` [454, L219, L233, L356, L415, L423, L424](#)
`\LoadClassWithOptions` [454, L232](#)
`\loccount` [N15](#)
`\log` [z3](#)
`\loggingall` [b454](#)
`\loggingoutput` [b450, b463, b479, b493](#)
`\Longleftarrow` [t400](#)
`\longleftarrow` [t397](#)
`\Longleftarrowrightarrow` [t406, t408](#)
`\longleftarrowrightarrow` [t404](#)
`\longmapsto` [t402](#)
`\Longrightarrow` [t394](#)
`\longrightarrow` [t395, t402](#)
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx
- `\loop` [a76, b387, C341, N153, N162](#)
`\lor` [t285](#)
`\lower` [j2, t366, B166, D15, D75, D162, D163, D200, D201](#)
`\lower@bound` [p340, p341, p352](#)
`\lowercase` [g26, l105, l884, o266, o324, y143, y150, O324](#)
`\lq` [b369](#)
`\lrbox` [B97](#)
`lrbox` (environment) [284](#)
`\ltx@sh@ft` [b438, l317, l324, l398, l406, l644, l651](#)
`\luabytecode` [N196](#)
`\luachunk` [N204](#)
`\luafunction` [N180](#)
`\luatexbase` [N253](#)
`\luatexluafunction` [a18, a23](#)
`\luatexversion` [a11, N5](#)
- ### M
- `\M` [b367](#)
`\m@ne` [b39](#)
`\m@th` [b418, b430, j13, t243, t367, t369, t370, t373, t414, t438, t441, t444, t447, t453, t456, t463, t466, t528, z68, z71, z89, z105, z108, z110, z115, z134, z259, z329, z341, z368, z378, B229, B329, C154, F159, G380, G385, G393, G403](#)
`\magstep` [b358](#)
`\magstephalf` [b358](#)
`\makeatletter` [d308, k26, o330, y19, F134, K2, L213, L318](#)
`\makeatother` [d308, L213, O390](#)
`\makebox` [284, z193, z219, B3](#)
`\makeglossary` [374, k69, H20](#)
`\makeindex` [374, k68, H3](#)
`\makelabel`
 . [A45, A97, A205, A218, A238, A249](#)
`\MakeLowercase` [O321, O330](#)
`\makeph@nt` [z84, z86](#)
`\MakeRobust` [d248](#)
`\makesm@sh` [z100, z102](#)
`\maketitle` [344](#)
`\MakeUppercase` [O314, O314](#)
`\mandatory@arg` [p368, p455, p459, p464, p471, p473, p478, p480, p485, p487, p498, p505, p507](#)
`\mapsto` [t352](#)
`\mapstochar` [t351, t352, t402](#)
`\marginpar` [G312](#)
`\marginparpush` [K92, K1744](#)
`\marginparsep` [K91, K1755, K1757](#)

- `\marginparwidth` . . . G341, K90, K1757
 - `\mark` J23, J31, J39
 - `\markboth` J18
 - `\markright` J18
 - `\marks` N37, O10, O12
 - `\math` z238
 - math (environment) z238
 - `\math@bgroup` o473, p260, p266, r53,
r81, r142, r172, v113, v114, v121
 - `\math@egroup`
o473, p264, p265, v114, v115, v122
 - `\math@fonts` o443, o448,
p186, p290, r60, r89, r149, r180
 - `\math@fontsfalse` . . . j7, l251, l278,
l291, l997, o42, o171, o181, o204
 - `\math@fontstrue` o169, o485
 - `\math@version` o8, o270, o447,
o451, o453, o454, o456, p184,
r56, r59, r64, r65, r69, r84, r88,
r93, r94, r98, r111, r112, r113,
r126, r127, r128, r145, r148,
r152, r154, r156, r160, r175,
r179, r183, r185, r187, r191, s67
 - `\mathaccent` r586, r614
 - `\mathalpha`
. r686, r847, t88, t89, t90, t91,
t92, t93, t94, t95, t96, t97, t98,
t99, t100, t101, t102, t103, t104,
t105, t106, t107, t108, t109,
t110, t111, t112, t113, t114,
t115, t116, t117, t118, t119,
t120, t121, t122, t123, t124,
t125, t126, t127, t128, t129,
t130, t131, t132, t133, t134,
t135, t136, t137, t138, t139,
t140, t141, t142, t143, t144,
t145, t146, t147, t148, t149,
t216, t217, t218, t219, t220,
t221, t222, t223, t224, t225,
t226, t424, t425, t426, t427,
t428, t429, t430, t431, t433, t436
 - `\mathbf` s14, t70
 - `\mathbin` r852,
t151, t152, t154, t276, t277,
t278, t279, t282, t284, t286,
t287, t288, t289, t290, t291,
t292, t293, t294, t295, t296,
t297, t298, t299, t300, t301,
t302, t303, t304, t305, t306,
t307, t308, t309, t310, t311, z37
 - `\mathcal` t69
 - `\mathchar`
b430, r629, r671, t228, t240, t521
 - `\mathchar@type` r614,
r659, r662, r671, r687, r787, r848
 - `\mathchardef`
b21, b22, b23, b24, b100, b103,
b104, e3, e4, e5, e6, l70, r662, N217
 - `\mathcharzero` N217
 - `\mathchoice` z61
 - `\mathclose` r855, t150,
t159, t161, t164, t169, t175,
t177, t179, t472, t497, t501,
t505, t509, t515, z43, z46, z49, z52
 - `\mathcode` r659, t171, t172, t173
 - `\mathdollar` l256, t518
 - `\mathellipsis` l270, t523
 - `\mathgroup` b76, l1204, o15, p257,
p263, p269, p270, p281, s82, t529
 - `\mathhexbox` b430, s92
 - `\mathindent` z318, z330, z342, z370, z380
 - `\mathinner` t412, t416, t421, t523
 - `\mathit` s29, t72, t75, t521
 - `\mathnormal` t68
 - `\mathop` r851,
t259, t260, t261, t262, t263,
t264, t265, t267, t268, t269,
t270, t271, t272, t274, t275,
t444, t447, z3, z4, z5, z6, z7, z8,
z9, z10, z11, z12, z13, z14, z15,
z16, z17, z18, z19, z20, z21, z22,
z23, z24, z25, z26, z27, z28, z29,
z30, z31, z32, z33, z34, z107, z246
 - `\mathopen` r854, t160, t163, t168, t174,
t176, t178, t470, t499, t503,
t507, t511, t513, z41, z44, z47, z50
 - `\mathord` r686,
r850, t155, t162, t165, t170,
t182, t183, t184, t186, t187,
t188, t189, t190, t191, t192,
t193, t194, t195, t196, t197,
t198, t199, t200, t201, t202,
t203, t204, t205, t206, t207,
t208, t209, t210, t211, t212,
t213, t214, t215, t227, t229,
t230, t231, t232, t233, t234,
t235, t236, t237, t238, t239,
t241, t242, t247, t248, t249,
t250, t252, t253, t254, t255,
t256, t257, t258, t432, t434,
t435, t455, t456, t459, t460,
t461, t462, t474, t476, t478,
t481, t495, t517, t518, t519, t520
 - `\mathpalette`
t365, t369, t372, z60, z69, z82, z98
 - `\mathparagraph` l259, m77, m89, t518
 - `\mathph@nt` z82, z88
- File Key:** a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

- `\mathpunct` r856, t153, t157, t409, t410, t411
- `\mathrel` r853, t156, t158, t166, t167, t180, t181, t244, t312, t313, t314, t315, t316, t317, t318, t319, t320, t321, t322, t323, t324, t325, t327, t329, t331, t332, t333, t334, t335, t336, t337, t338, t339, t340, t341, t343, t344, t345, t346, t347, t349, t351, t353, t354, t355, t356, t357, t358, t359, t360, t361, t362, t363, t365, t369, t372, t379, t381, t384, t385, t387, t390, t392, t483, t485, t487, t489, t491, t493, z42, z45, z48, z51, z107, z246
- `\mathring` t436
- `\mathrm` s5, t67
- `\mathsection` l260, m76, m88, t518
- `\mathsf` s8, t71, t74
- `\mathsm@sh` z98, z104
- `\mathsterling` l268, t518
- `\mathstrut` z94, z112, z113
- `\mathsurround` b418
- `\mathsymbol` r664
- `\mathtt` s11, t73
- `\mathunderscore` t518
- `\mathversion` o270, s64, s66
- `\matrix` z110, z114
- `\max` z22
- `\maxdeadcycles` K7
- `\maxdepth` b339, i183, k50, K99, K480, K488, K520, K625, K634, K674, K901, O79
- `\maxdimen` b288, b340, b341, b397, b435, b451, b462, b478, b493, o495, p338, p391, t366, D246, D291, D330, K265, K1763, K1783, K1788, K2094, K2095, K2097, O83
- `\maybe@ic` v46, v47, v66
- `\maybe@ic@` v66
- `\maybe@icfalse` v80
- `\maybe@ictrue` v70
- `\mb@b` B49, B59
- `\mb@l` B49, B53, B58, D47, D51
- `\mb@r` B49, B53, B58, D47, D51
- `\mb@t` B50, B57
- `\mbox` 284, b430, j13, l242, s88, t414, B11, B20, B23, D20, G385, G393, G403
- `\mddefault` s18, t32, t40
- `\mdseries` s16, s17, s91, v20
- 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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx
- `\meaning` a214, a223, a294, d205, d264, d317, r412, r425, r526, r586, r629, r722, r800, r904
- `\medbreak` b412
- `\medmuskip` t531, z36, z38, z145
- `\medskip` b415, i256
- `\medskipamount` b414, i257, i259
- `\MessageBreak` .. d181, d254, g3, g6, g13, g33, g46, g60, g73, g175, g177, g183, g190, l121, l889, l892, l916, l918, l919, l920, l922, l924, l925, l926, l927, l928, l977, l979, l987, l994, l1209, o391, o425, p20, p21, p67, p88, p281, p432, p452, p484, p497, p510, q31, q33, r367, r376, r514, v127, y23, K552, K1870, K1907, L94, L243, L254, L256, L258, L269, L325, L326, L328, L329, L330, L332, L334, L351, L352, L353, L354, L400, L417, L418, L450, L478, O216, O217, O218, O220
- `\mho` s104
- `\mid` t316
- `\min` z23
- `\minipage` B249
- `minipage (environment)` 285
- `\mit` s169
- `\mkern` . t228, t244, t246, t370, t379, t421, t422, t423, t451, t452, t453, t454, t455, t456, t457, t458, z36, z37, z40, z73, z74, F160
- `\models` t392
- `\module_error` 476, N309
- `\module_info` 476, N309
- `\module_warning` 476, N309
- `\modules` N262
- `\month` a180, c16, L462
- `\moveright` K596
- `\mp` t303
- `\mscount` C338
- `\mskip` i282, z36, z38, z144, z145, z146, z147
- `\mu` t198
- `\multicolumn` C194
- `\multipt` D25, D29
- `\multispan` C194, C338
- `\muskip` . b29, b55, b86, t451, t452, N34
- `\muskipdef` b55, b86, N218
- `\muskipzero` N218

N

`\n` N296, N298, N305, N307, N427, N506, N529, N560,

- N577, N599, N607, N608, N628,
N641, N648, N649, N656, N668
- \n@space . . . t524, t525, t526, t527, t528
- \nabla t239
- \narrower [b423](#)
- \natural t253
- \ncallback N565
- \ndefault N569
- \ne t326
- \narrow t319
- \NeedsTeXFormat p12, [L248](#), L521
- \neg t250, t251
- \negthinspace [i303](#)
- \neq t326
- \new@command
. d54, [d55](#), d108, d142, d161, d216
- \new@environment d123, [d124](#), d136
- \new@fontshape q2, q4, q22, [q24](#)
- \new@mathalphabet r409, r430, [r441](#)
- \new@mathgroup
. b75, b78, b91, b93, [o15](#), r289, N25
- \new@mathversion [r20](#), r246, [r248](#)
- \new@symbolfont r290, [r322](#)
- \new_attribute 474, [N375](#)
- \new_bytecode 474, [N406](#)
- \new_chunkname 474, [N417](#)
- \new_whatsit 474, [N395](#)
- \newattribute 473, [N77](#), N230
- \newbox [b47](#), b293,
b420, e13, z66, A27, B69, C16,
C17, C18, C302, D6, D355,
D360, K93, K127, K128, K129
- \newcatcodetable 473,
[N87](#), N96, N97, N123, N124, N234
- \newcommand
. 35, [d54](#), l4, t29, t30, t31, t32,
t33, t34, t35, t36, t37, t38, t39,
t40, t41, D367, K2180, K2183,
K2186, K2187, K2190, K2191
- \newcount [b47](#),
b296, b299, b301, b302, b303,
b304, b306, b308, b357, e7, e8,
i62, k9, m36, p25, r27, r254,
z55, z250, z251, A23, A24, A25,
A26, A56, A226, A241, B290,
C11, C12, C13, C14, C15, C294,
C295, C296, D349, D350, D351,
D352, D361, F19, F123, F124,
G3, G271, G272, G273, G274,
K110, K112, K114, K116, K118,
K126, K1896, K2178, K2181,
K2184, K2188, O3, O4, O5, O71
- \newcounter 132, [m10](#)
- File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- \newdimen [b47](#), b288, b290,
b291, b305, b356, e10, e11, e12,
i61, p352, p353, z53, z319, A9,
A10, A11, A12, A13, A14, A15,
A16, A17, A18, A19, A20, A21,
A22, B124, B125, C3, C5, C6,
C7, C8, C139, C297, C298,
C299, C300, D3, D4, D5, D7,
D216, D217, D218, D219, D220,
D221, D353, D354, D356, D357,
D358, D359, G404, K78, K79,
K80, K82, K83, K84, K85, K86,
K87, K88, K89, K90, K91, K92,
K98, K100, K101, K113, K115,
K117, K119, K120, K121, K122,
K123, K124, K125, K1897, K1898
- \newenvironment 36, [d123](#), L460
- \newfam b78, b93, o17, N38
- \newfont [s68](#)
- \newgroup [r47](#)
- \newhelp [b287](#)
- \newif [d145](#), e9,
k7, k8, l936, o169, r15, v65, x3,
z75, z76, z133, z252, A28, A29,
A30, A31, A32, A33, A138,
B304, C19, C212, D55, D212,
D213, D214, D215, D244, D245,
F21, F107, K102, K103, K104,
K105, K106, K107, K108, K109, L2
- \newinsert b186,
[b224](#), B291, G370, K27, K1762
- \newlabel [x22](#), x34
- \newlanguage [b47](#)
- \newlength 138, [n3](#)
- \newline [i43](#)
- \newlinechar a67, [d5](#)
- \newluabytecode 473, [N192](#), N244
- \newluachunkname 474, [N200](#), N246
- \newluafunction
. 473, N4, [N176](#), N228, N240
- \newmarks [O6](#)
- \newmathalphabet q13, [q109](#)
- \newmathalphabet@ q14
- \newmathalphabet@@ [q109](#)
- \newmathalphabet@@@ q15, [q109](#)
- \newmuskip [b47](#)
- \newpage K140, K146, [K157](#)
- \newread [b47](#), k3
- \newsavebox 284, [B69](#)
- \newskip [b47](#), b289,
b292, b354, b355, e14, e15, e17,
i259, i260, i261, i300, n3, y79,
z253, A2, A3, A4, A5, A6,
A7, A8, K2192, K2193, K2194,

- K2198, K2199, K2202, K2203,
 K2204, K2208, K2209, K2210
 \newtheorem [E1](#)
 \newtie [I732](#), [I1080](#), [I1081](#)
 \newtoks [b63](#),
 [b88](#), [b287](#), [e16](#), [o280](#), [o281](#), [p201](#)
 \newwhatsit [473](#), [N184](#), [N242](#)
 \newwrite [b47](#), [k4](#), [k5](#), [k6](#), [F137](#), [H4](#), [H21](#)
 \newXeTeXintercharclass [O21](#)
 \nfss@catcodes [o20](#), [o85](#),
 [o321](#), [o322](#), [o329](#), [t19](#), [t24](#), [t54](#), [K3](#)
 \nfss@text [l264](#), [l266](#), [s88](#), [v5](#), [v105](#), [x13](#)
 \NG [I421](#), [O328](#)
 \ng [I439](#), [O328](#)
 \ni [t341](#), [t342](#)
 \no@alphabet@error . [o5](#), [r268](#), [r270](#),
 [r446](#), [r447](#), [r461](#), [r470](#), [r556](#), [r557](#)
 \noaccents@ [o488](#), [t48](#)
 \noalign [t245](#),
 [t439](#), [t442](#), [t444](#), [t445](#), [t449](#),
 [t450](#), [z112](#), [z113](#), [z118](#), [z121](#),
 [z135](#), [z292](#), [C193](#), [C318](#), [C337](#), [D54](#)
 \noboundary [b307](#)
 \nobreak . . . [b400](#), [b403](#), [b405](#), [i38](#),
 [i53](#), [i79](#), [i93](#), [i119](#), [i243](#), [i251](#),
 [i270](#), [i277](#), [i298](#), [k67](#), [k79](#), [l359](#),
 [l361](#), [y69](#), [B367](#), [F73](#), [F157](#),
 [F158](#), [F162](#), [G440](#), [J25](#), [J33](#),
 [K310](#), [K1059](#), [K1225](#), [O135](#),
 [O137](#), [O141](#), [O142](#), [O143](#), [O147](#)
 \nobreakdashes [i262](#)
 \nobreakspace [i276](#)
 \nocite [376](#), [I39](#)
 \nocorr [v26](#), [v41](#), [v45](#), [v48](#)
 \nocorrlist [v72](#), [v104](#)
 \nofiles [81](#), [k63](#)
 \noindent [F122](#)
 \nointerlineskip
 [b395](#), [t245](#), [t439](#), [t442](#),
 [t445](#), [t449](#), [z192](#), [z218](#), [D271](#),
 [D274](#), [D315](#), [D317](#), [K1752](#), [K1760](#)
 \nolimits [t266](#), [t273](#),
 [z3](#), [z4](#), [z5](#), [z9](#), [z10](#), [z11](#), [z12](#), [z13](#),
 [z14](#), [z15](#), [z16](#), [z17](#), [z18](#), [z19](#), [z20](#),
 [z21](#), [z26](#), [z27](#), [z28](#), [z29](#), [z31](#), [z34](#)
 \nolinebreak [66](#), [i13](#)
 \non@alpherr [o467](#), [o469](#),
 [r72](#), [r101](#), [r117](#), [r163](#), [r194](#), [r926](#)
 \nonfrenchspacing . . . [b361](#), [b549](#), [k42](#)
 \nonscript [z36](#), [z38](#)
 \nonumber [z279](#), [z302](#), [z303](#)
 \nopagebreak [66](#), [i3](#)
 \normalbaselines . . . [b365](#), [z108](#), [z110](#)
 File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx
 \normalbaselineskip
 [b354](#), [b366](#), [p142](#), [B243](#)
 \normalcolor [z245](#), [z315](#), [B62](#),
 [B282](#), [F163](#), [G97](#), [G166](#), [K190](#),
 [K466](#), [K600](#), [K610](#), [K2115](#), [K2148](#)
 \normalfont [o496](#), [s93](#),
 [v18](#), [y120](#), [z245](#), [z315](#), [F163](#), [G381](#)
 \normallineskip . . . [b354](#), [b365](#), [B242](#)
 \normallineskiplimit [b354](#), [b366](#), [z136](#)
 \normalmarginpar [G367](#)
 \normalsfcodes [k38](#), [k40](#), [k42](#), [k62](#), [K589](#)
 \normalsize [k36](#),
 [v125](#), [G23](#), [G176](#), [G352](#), [K588](#), [L5](#)
 \not [t244](#), [t326](#), [t345](#)
 \not@base [s100](#),
 [s104](#), [s105](#), [s106](#), [s107](#), [s108](#),
 [s109](#), [s110](#), [s111](#), [s112](#), [s113](#), [s114](#)
 \not@math@alphabet [s5](#), [s8](#),
 [s11](#), [s14](#), [s17](#), [s20](#), [s23](#), [s26](#), [s29](#), [s47](#)
 \notin [t369](#)
 \nu [t199](#)
 \null [b379](#), [l276](#),
 [l294](#), [l408](#), [l411](#), [l715](#), [l718](#), [x17](#),
 [y108](#), [y132](#), [z91](#), [z110](#), [z128](#), [F157](#)
 \nulldelimiterspace [b343](#), [t528](#)
 \nullfont [y51](#)
 \number . . . [a81](#), [d2](#), [d91](#), [m51](#), [o451](#),
 [o454](#), [p393](#), [r64](#), [r93](#), [r113](#), [r128](#),
 [r153](#), [r184](#), [s85](#), [L431](#), [L462](#), [N108](#)
 \numberline [F55](#), [F65](#), [F166](#), [G17](#)
 \numexpr [b182](#), [b198](#),
 [b208](#), [b228](#), [K36](#), [N85](#), [N108](#), [N160](#)
 \nunknown [N582](#)
 \nwarrow [t321](#)
- O**
- \O [I194](#), [I327](#), [I423](#), [l654](#), [O327](#)
 \o [I203](#), [I332](#), [I441](#), [l662](#), [O327](#)
 \o@lign [b432](#),
 [l317](#), [l324](#), [l398](#), [l406](#), [l644](#), [l651](#)
 \oalign [b432](#)
 \obeycr [i309](#)
 \obeylines [b382](#), [y114](#), [y127](#), [y128](#), [K557](#)
 \obeyspaces [b382](#), [K557](#)
 \oddsidemargin [K79](#), [K81](#), [K582](#)
 \odot [t298](#)
 \OE [I193](#), [I326](#), [I422](#), [l653](#), [O327](#)
 \oe [I202](#), [I331](#), [I440](#), [l663](#), [O327](#)
 \of [z67](#), [z249](#)
 \offinterlineskip [b395](#)
 \oint [t273](#)
 \ointop [t272](#), [t273](#)
 \oldstylenums [I1201](#), [s78](#)
 \Omega [t226](#)

`\omega` t209
`\ominus` t301
`\omit` z121, z122, C328, C331, C338, C342
`\on@line` g8, g15, g165, y56, B103, L350
`\onecolumn` K148
`\OnlyDescription` p5, u3
`\oalign` b432, l276, l288,
l321, l402, l408, l410, l615, l648,
l715, l718, l764, s90, t370, t373
`\openup` z129, z134
`\operator@font`
..... t529, z3, z4, z5, z6, z7,
z8, z9, z10, z11, z12, z13, z14,
z15, z16, z17, z18, z19, z20, z21,
z22, z23, z24, z25, z26, z27, z28,
z29, z30, z31, z32, z33, z34, z37, z40
`\oplus` t302
`\optional@arg`
... p369, p448, p450, p504, p507
`\OptionNotUsed` L143, L150, L365
`\Orb` l598
`\oslash` t299
`\otimes` t300
`\outer` b11, N19, N38
`\outer@nobreak`
... G181, G251, G256, G259, G346
`\outerparskip` A1
`\output` K230
`\outputpenalty` K232,
K246, K269, K272, K273, K308,
K1069, K1070, K1235, K1238
`\oval` D235, D238
`\over` t377, z107, z247
`\overbrace` t444
`\overfullrule` b338, J55
`\overleftarrow` t441
`\overrightarrow` t438
`\owns` t342

P

`\P` l259
`\p@` b290
`\p@equation` z257, z377
`\p@reset@font` s93
`\p@selectfont` p117
`\PackageError` g84, l887, l942, l986
`\PackageInfo`
... g84, l916, l932, l933, l993, l1278
`\PackageWarning` g84, l943, l1207
`\PackageWarningNoLine` ... g84, K1869
`\pagebreak` 66, i3
`\pagegoal` K1790, K1797
`\pagenumbering` 244, w5
`\pageref` x10
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

`\pageshrink` K512, K516, K532
`\pagestyle` J2
`\pagetotal` K135
`\paperheight` K100
`\paperwidth` K100
`\par` a115, b11, b375, b383,
b384, b399, b408, b409, b410,
b412, b414, b416, d6, h3, h4,
h6, y49, y69, y106, A63, A110,
A127, A129, A135, A161, A164,
B234, B278, C168, C344, F24,
F73, F164, G15, G24, G249,
J48, J49, K169, K231, K1796, N159
`\par@deathcycles` A56, A77, A79, A80
`\paragraphmark` F126
`\parallel` t315
`\parbox` 284, B186
`\parboxrestore` B245
`\parfillskip` b353, o495,
y78, y91, y103, A76, B242, F152
`\parindent` . b345, b424, b425, y78,
y85, y91, y103, A50, B237, F153
`\parsep` A1, A49, A90
`\parseunicodedataI` N126, N165
`\parseunicodedataII` N127, N129
`\parseunicodedataIII` N131, N137
`\parseunicodedataIV` N133, N145
`\parseunicodedataV` N149, N152
`\parshape` A54
`\parskip` b346, y70,
y101, y103, z386, A49, A73,
A88, A90, A117, A153, A172,
A223, B237, C68, K1069, K1237
`\partial` t235
`\partopsep` z384, A1, A61
`\PassOptionsToClass` 454, L120
`\PassOptionsToPackage` 454, L120
`\patch@level` c1, c35, c40, c42,
c44, c47, c49, O338, O350, O352
`\patterns` l155
`\penalty`
b404, b405, b406, b407, b408,
b409, b413, b415, b417, i7,
i10, i21, i177, i187, i212, i216,
v101, y108, y111, z37, z137,
z292, A190, C56, G195, G199,
G201, G217, G221, G223, K143,
K171, K172, K1067, K1233, I17
`\perp` t355
`\ph@nt` z77, z78, z79, z80
`\phantom` z75
`\Phi` t224
`\phi` t206
`\Pi` t221

- \pi t201
 - \pickup@font l131, o160, o287,
o402, o436, p122, p285, p287, p289
 - \pictur@ D8
 - \picture D8
 - \pm t304
 - \pmatrix z114
 - \pmod z39
 - \poptabs g210, C127
 - \poptracing p130, p294
 - \postdisplaypenalty
..... i28, z327, z339, z365
 - \pounds l267
 - \Pr z32
 - \pr@@s z156, z164
 - \pr@@@t z159, z165
 - \pr@m@s z153, z154
 - \prec t332
 - \preceq t335
 - \predisplaypenalty
..... b324, z326, z338, z364
 - \preload@sizes q11, q94
 - \pretolerance b311, o497
 - \prevdepth b395, b399,
b400, i183, i184, i241, i246,
z135, G196, G198, G218, G220
 - \prim@s z150, z152, z164
 - \prime t172, t237, z153
 - \prime@s z151
 - \process@table k34, r200
 - \ProcessOptions
..... l907, l945, p71, L151, L194, L419
 - \ProcessOptions* L151
 - \prod t267
 - \propto t312
 - \protect d79, d196, d211, d220, d225,
d228, d229, d231, d232, d237,
d238, d243, d246, d247, d269,
g201, g203, g204, g210, g216,
g223, g231, g234, g240, k75, l26,
l32, l51, l55, l159, l167, r475,
r931, s71, v126, x12, C225, F11,
F55, F65, F143, G17, K566, I5
 - \protected m103
 - \protected@edef
..... d230, m101, x37, B298, F43,
G420, O318, O324, O329, O330
 - \protected@write
..... k66, k71, x33, F145, H14, H31
 - \protected@xdef
..... d230, F10, G406, G430, G446
 - \provide@command d155, d156
 - \providecommand d155, l6, K1880
 - \provides_module 476, N263
 - File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
 - \ProvidesClass 453, L107
 - \ProvidesFile
..... a84, t551, t553, t554, t555, L109
 - \ProvidesPackage
..... 454, p13, L90, L107, L522
 - \ProvideTextCommand l3, l60
 - \ProvideTextCommandDefault l57
 - \ps@empty J10, O85
 - \ps@plain J13
 - \Psi t225
 - \psi t208
 - \pushtabs g210, C124
 - \pushtracing p115, p275
 - \put D21, D176, D177, D178,
D179, D184, D186, D198, D199,
D200, D201, D206, D209, D404
- Q**
- \qbezier 318, D367
 - \qbeziermax D366, D388
 - \quad i306
 - \quad i306, z109, z111, z120, F94
 - \quotedblbase l442, l664
 - \quotesinglbase l443
- R**
- \r b373, b374, l186, l315, l355, l392,
l494, l521, l531, l557, l640, l676
 - \r@t z66
 - \radical r797, r800, r830
 - \raggedbottom J39
 - \raggedleft y86, y88
 - \raggedright y80, y82
 - \raise l276,
l290, l354, l357, l616, l678, l765,
s91, t373, t421, t423, z73, B352,
B361, D22, D32, D74, D162,
D237, D265, D309, D337, D415
 - \raisebox 285, B330
 - \rangle t496
 - \rbrace l258, t500
 - \rbrack b371
 - \rceil t504
 - \Re t233
 - \ref x10
 - \refstepcounter l32, x32,
z243, z366, A202, E27, F42, G9
 - \registernumber 474, N354
 - \Relbar t384, t392, t394, t400
 - \relbar t381, t396, t398
 - \relpenalty b319
 - \rem@pt o263
 - \remove@angles p301, p324
 - \remove@nil r36

\remove@star [p301](#), [p307](#)
 \remove@to@nnil [o262](#), [p301](#), [p327](#), [p440](#)
 \remove_from_callback .. [476](#), [N639](#)
 \removelastskip [b411](#), [b413](#), [b415](#), [b417](#)
 \renew@command [d101](#), [d102](#), [d162](#), [d170](#)
 \renew@environment [d129](#), [d130](#)
 \renewcommand [36](#), [d101](#), [z314](#), [z334](#), [z355](#)
 \renewenvironment [36](#), [d129](#), [z363](#), [z375](#)
 \repeat [a76](#), [a78](#), [b387](#), [C341](#), [N157](#), [N167](#)
 \RequirePackage [454](#), [K1877](#),
 [L209](#), [L216](#), [L237](#), [L415](#), [N22](#)
 \RequirePackageWithOptions [454](#), [L235](#)
 \reserved@a [a116](#),
 [a120](#), [a121](#), [a190](#), [a191](#), [a194](#),
 [a212](#), [a216](#), [a238](#), [a245](#), [a248](#),
 [a250](#), [a251](#), [a258](#), [a261](#), [a263](#),
 [a264](#), [a271](#), [a274](#), [a276](#), [a302](#),
 [a303](#), [a304](#), [b186](#), [c12](#), [c18](#), [c33](#),
 [d94](#), [d97](#), [d110](#), [d111](#), [d112](#),
 [d114](#), [d161](#), [d162](#), [d163](#), [d169](#),
 [d170](#), [d171](#), [d172](#), [d175](#), [d194](#),
 [d203](#), [d207](#), [d262](#), [d266](#), [d291](#),
 [d300](#), [f33](#), [f37](#), [g189](#), [i265](#), [i268](#),
 [k76](#), [k77](#), [k99](#), [k100](#), [k138](#), [k140](#),
 [k145](#), [k147](#), [k149](#), [k155](#), [k159](#),
 [k167](#), [k170](#), [k183](#), [k184](#), [k188](#),
 [k194](#), [k213](#), [k217](#), [k221](#), [l75](#),
 [l77](#), [l85](#), [l102](#), [l107](#), [o30](#), [o33](#),
 [o36](#), [o70](#), [o73](#), [o75](#), [o112](#), [o116](#),
 [o323](#), [o326](#), [o374](#), [o375](#), [o390](#),
 [o393](#), [o398](#), [o409](#), [o410](#), [o423](#),
 [o427](#), [o432](#), [o459](#), [o462](#), [o463](#),
 [o471](#), [p150](#), [p152](#), [p154](#), [p164](#),
 [p166](#), [p169](#), [p298](#), [p299](#), [p312](#),
 [p313](#), [q53](#), [q57](#), [r356](#), [r365](#), [r367](#),
 [r411](#), [r414](#), [r424](#), [r427](#), [r525](#),
 [r527](#), [r585](#), [r587](#), [r628](#), [r630](#),
 [r721](#), [r723](#), [r799](#), [r801](#), [r903](#),
 [r905](#), [r921](#), [r923](#), [r924](#), [r929](#),
 [v30](#), [v31](#), [v36](#), [v37](#), [v48](#), [v51](#),
 [v71](#), [v78](#), [y41](#), [y42](#), [y54](#), [y55](#),
 [y59](#), [y64](#), [y65](#), [z294](#), [z295](#), [z296](#),
 [z297](#), [z299](#), [B51](#), [B52](#), [B55](#), [B98](#),
 [B104](#), [C202](#), [C206](#), [C211](#), [C230](#),
 [C319](#), [C320](#), [D78](#), [D80](#), [D84](#),
 [D249](#), [D293](#), [D294](#), [G29](#), [G30](#),
 [G32](#), [G33](#), [G63](#), [G67](#), [G72](#), [G74](#),
 [G76](#), [G78](#), [G83](#), [G84](#), [G132](#),
 [G136](#), [G142](#), [G145](#), [G148](#),
 [G151](#), [K37](#), [K46](#), [K48](#), [K50](#),
 [K787](#), [K807](#), [K1873](#), [K1875](#),
 [K1876](#), [K1965](#), [K1967](#), [K1973](#),
 [K1976](#), [L77](#), [L84](#), [L88](#), [L202](#),
 [L205](#), [L249](#), [L250](#), [L253](#), [L290](#),
 [L294](#), [L306](#), [L307](#), [L309](#), [L319](#),
 [L359](#), [L524](#), [L526](#), [O153](#), [O170](#),
 [O172](#), [O173](#), [O181](#), [O183](#), [O184](#),
 [O228](#), [O259](#), [O265](#), [O266](#), [O268](#),
 [O270](#), [O274](#), [O286](#), [O288](#), [O289](#),
 [O297](#), [O299](#), [O300](#), [O316](#), [O317](#),
 [O318](#), [O319](#), [O322](#), [O323](#), [O324](#),
 [O325](#), [O351](#), [O354](#), [O355](#), [O372](#)
 \reserved@b [a117](#), [a118](#),
 [d86](#), [d88](#), [d95](#), [d112](#), [d113](#), [d204](#),
 [d205](#), [d207](#), [d263](#), [d264](#), [d266](#),
 [d292](#), [d302](#), [f33](#), [f34](#), [f37](#), [i266](#),
 [i267](#), [i274](#), [k98](#), [k100](#), [k150](#),
 [k152](#), [k154](#), [k216](#), [k222](#), [l78](#),
 [l85](#), [o60](#), [o62](#), [o115](#), [o116](#), [o460](#),
 [o471](#), [q47](#), [q54](#), [q71](#), [q73](#), [r282](#),
 [r284](#), [r337](#), [r339](#), [r364](#), [r365](#),
 [r366](#), [r401](#), [r403](#), [r482](#), [r484](#),
 [r529](#), [r530](#), [r531](#), [r538](#), [v35](#), [v36](#),
 [v49](#), [v51](#), [v78](#), [v79](#), [C207](#), [C209](#),
 [C211](#), [G43](#), [G44](#), [G112](#), [G113](#),
 [K696](#), [K699](#), [K713](#), [K716](#), [K733](#),
 [K736](#), [L78](#), [L79](#), [L81](#), [L298](#),
 [L304](#), [L307](#), [L467](#), [L468](#), [L470](#),
 [L496](#), [O156](#), [O158](#), [O162](#), [O231](#),
 [O233](#), [O237](#), [O317](#), [O323](#), [O372](#)
 \reserved@c [a118](#), [a123](#),
 [d297](#), [d300](#), [d302](#), [d305](#), [k205](#),
 [k206](#), [o61](#), [o62](#), [o461](#), [o464](#), [q48](#),
 [q55](#), [q61](#), [q68](#), [r33](#), [r37](#), [r283](#),
 [r284](#), [r338](#), [r339](#), [r402](#), [r403](#),
 [r483](#), [r484](#), [r506](#), [r515](#), [r530](#),
 [r544](#), [r711](#), [r728](#), [r737](#), [r765](#),
 [r776](#), [r816](#), [r829](#), [r831](#), [v50](#), [v52](#),
 [v59](#), [L444](#), [L445](#), [L446](#), [L456](#),
 [L472](#), [L479](#), [L504](#), [O160](#), [O165](#),
 [O175](#), [O235](#), [O256](#), [O257](#), [O258](#),
 [O260](#), [O261](#), [O262](#), [O263](#), [O264](#),
 [O267](#), [O269](#), [O281](#), [O291](#), [O374](#)
 \reserved@d [a121](#), [a124](#),
 [d290](#), [d299](#), [k204](#), [k206](#), [q61](#),
 [q68](#), [q70](#), [q74](#), [r719](#), [r728](#), [r737](#),
 [r773](#), [r776](#), [r824](#), [r829](#), [r833](#), [O375](#)
 \reserved@e .. [i36](#), [i38](#), [i47](#), [i53](#), [q39](#),
 [q45](#), [q70](#), [q73](#), [q74](#), [r34](#), [r39](#), [O376](#)
 \reserved@f
 . [i37](#), [i38](#), [i53](#), [l883](#), [l884](#), [l885](#),
 [l886](#), [l888](#), [l895](#), [o155](#), [o157](#),
 [o163](#), [o164](#), [p336](#), [p347](#), [p351](#),
 [p355](#), [p361](#), [p364](#), [p403](#), [p440](#),
 [p443](#), [q27](#), [q38](#), [q45](#), [q71](#), [q73](#), [O377](#)
 \reset@font [s93](#), [x13](#), [B295](#),
 [G175](#), [G351](#), [G415](#), [J14](#), [K587](#), [I20](#)
 \restgbl@settings [p222](#), [p232](#)

File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
 e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
 j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
 o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx

`\restore@mathversion`
 [r107](#), [r110](#), [r125](#), [r133](#)
`\restore@protect` [d230](#)
`\restorecr` [i309](#)
`\reversemarginpar` [G367](#)
`\rfloor` [t508](#)
`\rgroup` [t512](#)
`\rhd` [s113](#)
`\rho` [t202](#)
`\rhook` [t387](#), [t388](#)
`\right` [t524](#),
 [t525](#), [t526](#), [t527](#), [z109](#), [z114](#), [z127](#)
`\Rightarrow` [t325](#), [t394](#), [t406](#)
`\rightarrow` [t349](#),
 [t350](#), [t352](#), [t386](#), [t396](#), [t404](#), [t455](#)
`\rightarrowfill` [t439](#), [t453](#)
`\rightharpoonowdown` [t363](#)
`\rightharpoonup` [t362](#), [t374](#)
`\rightthyphenmin` [b302](#), [M11](#)
`\rightleftharpoons` [t372](#)
`\rightline` [B368](#)
`\rightmargin` [A9](#), [A40](#), [A51](#)
`\rightmark` [J34](#)
`\rightskip` [b425](#), [y77](#),
 [y83](#), [y90](#), [y102](#), [A75](#), [B241](#), [F152](#)
`\rlap` [l354](#),
 [l357](#), [l678](#), [z304](#), [z315](#), [B372](#), [C70](#)
`\rlh@` [t372](#), [t373](#)
`\rmdefault` [s6](#), [s81](#), [t29](#), [t39](#)
`\rmfamily` [s4](#), [s5](#), [v15](#)
`\rmoustache` [t471](#)
`\Roman` [132](#), [m47](#)
`\roman` [132](#), [m46](#)
`\romannumeral`
 [m52](#), [m53](#), [A43](#), [A234](#), [A245](#)
`\root` [z66](#), [z249](#)
`\rootbox` [z66](#)
`\rq` [b369](#)
`\rule` [285](#), [B302](#), [B305](#), [G425](#)

S

`\S` [l260](#)
`\s@fct@` [p380](#), [p444](#)
`\s@fct@fixed` [p501](#)
`\s@fct@gen` [p456](#)
`\s@fct@genb` [p461](#)
`\s@fct@sgen` [p456](#)
`\s@fct@sgenb` [p461](#)
`\s@fct@sub` [p468](#)
`\s@fct@subf` [p493](#)
`\samepage` [66](#), [i27](#)
`\savebox` [284](#), [B70](#)
`\savecatcodetable` . [N120](#), [N171](#), [N173](#)

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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
 t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
 y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
 D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
 I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
 N=ltluatex.dtx, O=ltfinal.dtx

- `\SetMathAlphabet`
 - ... [o12](#), [q140](#), [q141](#), [r480](#), [t74](#), [t75](#)
- `\SetMathAlphabet@` ... [r418](#), [r487](#), [r496](#)
- `\setminusminus` [t307](#)
- `\setrangeatcode`
 - [N99](#), [N107](#), [N116](#), [N117](#)
- `\SetSymbolFont` [r335](#), [t64](#), [t65](#), [t66](#)
- `\SetSymbolFont@` [r308](#), [r342](#), [r350](#)
- `\settodepth` [138](#), [n17](#)
- `\settoheight` [138](#), [n17](#)
- `\settowidth` [138](#), [n17](#)
- `\sf@size` [j6](#), [l251](#), [o189](#), [o208](#), [o483](#),
 - [p282](#), [p286](#), [G385](#), [G393](#), [G403](#)
- `\sfcode` [b361](#), [b362](#), [b363](#),
 - [b364](#), [b448](#), [i272](#), [k39](#), [O179](#), [O295](#)
- `\sfdefault` [s9](#), [t29](#)
- `\sffamily` [s7](#), [s8](#), [v16](#)
- `\sh@ft` [b436](#)
- `\shapedefault` [r240](#), [s97](#), [t38](#)
- `\sharp` [t254](#)
- `\shipout` [K571](#)
- `\shortstack` [D42](#)
- `\showboxbreadth`
 - ... [b333](#), [b451](#), [b504](#), [b521](#), [b537](#)
- `\showboxdepth`
 - [b334](#), [b451](#), [b503](#), [b520](#), [b538](#), [o497](#)
- `\showhyphens` [o491](#)
- `\showoutput` [b450](#)
- `\showoverfull` . [b449](#), [b452](#), [b486](#), [b494](#)
- `\Sigma` [t222](#)
- `\sigma` [t203](#)
- `\sim` [t353](#), [t365](#)
- `\simeq` [t354](#)
- `\sin` [z9](#)
- `\sinh` [z11](#)
- `\sixt@cn` .. [a66](#), [b16](#), [b64](#), [b66](#), [b89](#),
 - [b90](#), [b91](#), [o15](#), [r84](#), [r175](#), [r580](#),
[r582](#), [r623](#), [r625](#), [r667](#), [r669](#),
[r707](#), [r709](#), [r715](#), [r717](#), [r761](#),
[r763](#), [r769](#), [r771](#), [r812](#), [r814](#),
[r820](#), [r822](#), [D135](#), [D150](#), [D152](#),
[G62](#), [G80](#), [G131](#), [G153](#), [K915](#),
[K961](#), [K1100](#), [K1268](#), [K1502](#),
[K1566](#), [K1623](#), [K1693](#), [K1919](#),
[K1928](#), [K1984](#), [K2000](#), [K2033](#), [N30](#)
- `\size@update` .. [p128](#), [p139](#), [p158](#), [p160](#)
- `\sizefn@info`
 - ... [p306](#), [p308](#), [p316](#), [p344](#), [p358](#)
- `\skew` [t451](#)
- `\skip` [b28](#), [b53](#), [b85](#), [b201](#), [b232](#), [b276](#),
 - [B281](#), [G371](#), [K290](#), [K464](#), [N33](#)
- `\skip@` [b41](#),
 - [b398](#), [b400](#), [b401](#), [b403](#), [v88](#), [v91](#)
- `\skipdef` [b45](#), [b53](#), [b85](#), [N219](#)
- 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=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- `\skipzero` [N219](#)
- `\slash` [b404](#)
- `\sldefault` [s24](#), [t34](#)
- `\sloppy` [B244](#), [J43](#), [J48](#)
- `\sloppypar` [J48](#)
- `sloppypar (environment)` [J48](#)
- `\slshape` [l364](#), [l681](#), [s22](#), [s23](#), [v22](#)
- `\smallbreak` [b412](#)
- `\smallint` [t275](#)
- `\smallskip` [b413](#), [i256](#)
- `\smallskipamount` [b412](#), [i256](#), [i259](#)
- `\smash` . [t381](#), [t453](#), [t454](#), [t457](#), [t458](#), [z95](#)
- `\smile` [t358](#)
- `\sp` [z142](#)
- `\sp@n` [C338](#)
- `\space` [b377](#)
- `\spacefactor` [b402](#), [b403](#), [i67](#),
 - [i76](#), [i91](#), [i103](#), [i117](#), [i131](#), [i272](#),
[i288](#), [i293](#), [l70](#), [l71](#), [G440](#), [G442](#)
- `\spaceskip` [s80](#)
- `\spadesuit` [t258](#)
- `\span` [C342](#)
- `\split@name` [o291](#), [o303](#), [o354](#), [p473](#), [p487](#)
- `\splitfirstmark` [K2100](#)
- `\splitmaxdepth` ... [b340](#), [G418](#), [K2094](#)
- `\splittopskip` [b352](#), [G417](#)
- `\sqcap` [t290](#)
- `\sqcup` [t291](#)
- `\sqrt` [z248](#)
- `\sqrtsign` [t437](#), [z71](#), [z248](#)
- `\sqsubset` [s109](#)
- `\sqsubseteq` [t313](#)
- `\sqsupset` [s110](#)
- `\sqsupseteq` [t314](#)
- `\SS` [l253](#), [l424](#), [O328](#)
- `\ss` [l204](#), [l333](#), [l444](#), [l665](#), [O328](#)
- `\ssf@size` [l277](#),
 - [l291](#), [o190](#), [o209](#), [o484](#), [p282](#), [p288](#)
- `\stackrel` [z246](#)
- `\star` [t311](#)
- `\stepcounter`
 - [132](#), [m17](#), [m27](#), [o456](#), [r48](#), [x36](#),
[z256](#), [z299](#), [z376](#), [G405](#), [G429](#), [K617](#)
- `\stop` [y49](#)
- `\storedpar` [N159](#), [N164](#)
- `\stretch` [i302](#)
- `\strip@prefix` [a106](#),
 - [a223](#), [a294](#), [d205](#), [d264](#), [d316](#), [o441](#)
- `\strip@pt` [b440](#),
 - [o181](#), [o187](#), [o188](#), [o189](#), [o190](#),
[o203](#), [o207](#), [o263](#), [o483](#), [o484](#), [p134](#)
- `\strut` [b420](#), [z121](#), [z122](#), [C29](#)
- `\strutbox` [b420](#), [p143](#),
 - [B302](#), [C159](#), [C160](#), [G418](#), [G425](#)

`\sub@sfcnt` p468, p469, p470
`\subf@sfcnt` p493, p494, p495
`\subparagraphmark` F126
`\subsectionmark` F126
`\subset` t337
`\subseteq` t339
`\subst@correction` o50, o56
`\subst@fontshape` q8, q80
`\subst@size` p419
`\subsubsectionmark` F126
`\succ` t331
`\succeq` t334
`\sum` t268
`\sup` z24
`\suppressfloats` K1882
`\supset` t336
`\supseteq` t338
`\surd` t240
`\sw@slant` v74, v84
`\swarrow` t322
`\symbol` l122, s68
`\symletters` l1204, s82
`\symoperators` t529

T

`\T` g23, L505, L509, L510
`\t` .. l232, l625, l730, l924, l1191, l1193
`\t@st@ic` v73, v77
`\tabbing` C60
`\tabbingsep` C119, C121, C139
`\tabcolsep` C220, C297
`\tabskip` b431, z138,
z139, z261, z264, z267, z269,
z380, z393, z396, z398, C140, C165
`\tabular` C147
`\tabular*` C148
`\tabularnewline` C167, C180
`\tan` z15
`\tanh` z17
`\tau` t204
`\tc@check@accent`
..... l1003, l1079, l1081, l1083
`\tc@check@symbol` l1003,
l1073, l1075, l1077, l1085, l1087,
l1089, l1091, l1093, l1095, l1097,
l1099, l1101, l1103, l1105, l1107,
l1109, l1111, l1113, l1115, l1117,
l1119, l1121, l1123, l1125, l1127,
l1129, l1131, l1133, l1135, l1137,
l1139, l1141, l1143, l1145, l1147,
l1149, l1151, l1153, l1155, l1157,
l1159, l1161, l1163, l1165, l1167,
l1169, l1171, l1173, l1175, l1177,
l1179, l1181, l1183, l1185, l1187
`\tc@error` l983, l1004
`\tc@errorwarn` l942, l943, l976
`\tc@fake@euro` l991, l1072
`\tc@forcedfalse` l936
`\tc@forcedtrue` l941
`\tc@subst` l975, l975, l1003
`\tencirc` u10, D37, D363
`\tencircw` u10, D39
`\tenln` u9, D37, D38, D362, D364
`\tenlnw` u9, D39, D40
`\TeX` j1, j12, N225
`\TeXOrMath` m70, m86
`\text@command` v8, v29
`\textacutedbl` l785, l1036
`\textascendercompwordmark` l735, l1019
`\textasciicircum` l835, l1060
`\textasciibreve` l783, l1033
`\textasciicaron` l784, l1034
`\textasciicircum` l236, l445
`\textasciidieresis` l823, l1050
`\textasciigrave` l774, l1031
`\textasciimacron` l830, l1055
`\textasciitilde` l237, l446
`\textasteriskcentered`
. l217, l601, l745, l1026, m73, m79
`\textbackslash` l218, l447, l602
`\textbaht` l809, l1166, l1167
`\textbar` l219, l448, l603
`\textbardbl` l220, l604, l789, l1039, m78
`\textbf` v19
`\textbigcircle` l613, l762, l1118, l1119
`\textblank` l742, l1088, l1089
`\textborn` l775, l1124, l1125
`\textbraceleft` .. l221, l257, l449, l605
`\textbraceright` . l222, l258, l450, l606
`\textbrokenbar` l821, l1048
`\textbullet` l223, l607, l791, l1041
`\textcapitalcompwordmark` . l734, l1018
`\textcelsius` l792, l1042
`\textcent` l817, l1045
`\textcentoldstyle` . l794, l1140, l1141
`\textcircled` l229, l233, l249,
l250, l614, l763, l923, l1188, l1190
`\textcircledP` l828, l1182, l1183
`\textcolonmonetary` . l796, l1142, l1143
`\textcommaabove`
. l284, l286, l300, l301, l382, l585
`\textcommabelow` l273,
l275, l281, l282, l584, l586, l587,
l588, l589, l590, l591, l592, l593
`\textcompsubstdefault` l980, l982
`\textcompwordmark` l238, l451
`\textcopyleft` l826, l1180, l1181
`\textcopyright` . l233, l266, l824, l1051

File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx

`\textcurrency` 1819, 1918, 1922, 11076, 11077
`\textdagger` 1225, 1261, 1609, 1787, 11037, m74, m80
`\textdaggerdbl` 1224, 1262, 1608, 1788, 11038, m75, m81
`\textdblhyphen` 1746, 11090, 11091
`\textdblhyphenchar` . 1782, 11136, 11137
`\textdegree` 1831, 11056
`\textdied` 1777, 11128, 11129
`\textdiscount` 1811, 11170, 11171
`\textdiv` 1848, 11070
`\textdivorced` 1776, 11126, 11127
`\textdollar` 1205, 1256, 1362, 1452, 1679, 1743, 11024, 11196, 11198
`\textdollaroldstyle` 1793, 11138, 11139
`\textdong` 1805, 11160, 11161
`\textdownarrow` 1773, 11122, 11123
`\texteightoldstyle` . 1756, 11108, 11109
`\textellipsis` 1245, 1270
`\textemdash` 1206, 1334, 1453, 1666
`\textendash` 1207, 1335, 1454, 1667
`\textestimated` 1812, 1921, 11074, 11075
`\texteuro` 1846, 1919, 11071, 11072
`\textexclamdown` 1208, 1336, 1338, 1455, 1668
`\textfiveoldstyle` . 1753, 11102, 11103
`\textfloatsep` K638, K651, K2014, K2064, K2192
`\textflorin` 1795, 11043
`\textfont` p291, z148
`\textfont@name` p285, p291
`\textfouroldstyle` . 1752, 11100, 11101
`\textfraction` K1827, K1830, K1854, K1857, K2006, K2186
`\textfractionsolidus` 1747, 11027
`\textgravedbl` 1786, 11035
`\textgreater` 1231, 1456, 1624
`\textguarani` 1799, 11148, 11149
`\textheight` k16, k17, G261, G262, G265, G291, G305, K85, K199, K200, K248, K373, K421, K448, K616, K673, K725, O83, O84
`\texthyphen` 1210, 1341, 1458, 1670
`\texthyphenchar` . 1209, 1340, 1457, 1669
`\textinterrobang` .. 1803, 11156, 11157
`\textinterrobangdown` 1804, 11158, 11159
`\textit` v21
`\textlangle` 1758, 11112, 11113
`\textlbrackdbl` 1770, 11029
`\textleaf` 1778, 11130, 11131
`\textleftarrow` 1740, 11084, 11085
`\textless` 1230, 1459, 1623
`\textlira` 1801, 11152, 11153
`\textlnot` 1827, 11053
`\textlquill` 1815, 11176, 11177
`\textmarried` 1779, 11132, 11133
`\textmd` v19
`\textmho` 1761, 11116, 11117
`\textminus` 1759, 11028
`\textmu` 1836, 11061
`\textmusicalnote` .. 1780, 11134, 11135
`\textnaira` 1798, 11146, 11147
`\textnineoldstyle` . 1757, 11110, 11111
`\textnormal` v15
`\textnumero` 1810, 11168, 11169
`\textogonekcentered` .. 1409, 1582, 1583
`\textohm` 1769, 1920, 11073
`\textonehalf` 1844, 11067
`\textoneoldstyle` .. 1749, 11094, 11095
`\textonequarter` 1843, 11066
`\textonesuperior` 1840, 11064
`\textopenbullet` ... 1813, 11172, 11173
`\textordfeminine` ... 1254, 1825, 11052
`\textordmasculine` .. 1255, 1841, 11065
`\TextOrMath` .. m73, m74, m75, m76, m77, m78, m79, m80, m81, m93
`\textparagraph` 1226, 1259, 1610, 1837, 11062, m77
`\textperiodcentered` 1227, 1611, 1838, 11063
`\textpertenthousand` 1414, 1807, 11162, 11163, 11200
`\textperthousand` 1412, 1790, 11040, 11199
`\textpeso` 1800, 11150, 11151
`\textpilcrow` 1808, 11164, 11165
`\textpm` 1832, 11057
`\textquestiondown` 1211, 1337, 1339, 1460, 1671
`\textquotedbl` 1463
`\textquotedblleft` 1212, 1342, 1461, 1672
`\textquotedblright` 1213, 1343, 1462, 1673
`\textquoteleft` .. 1214, 1344, 1464, 1674
`\textquoteright` . 1215, 1345, 1465, 1675
`\textquotesingle` 1744, 11025
`\textquotestraightbase` .. 1736, 11020
`\textquotestraightdblbase` 1737, 11021
`\textrightangle` 1760, 11114, 11115
`\texttrbrackdbl` 1771, 11030
`\textrecipe` 1802, 11154, 11155
`\textreferencemark` . 1839, 11184, 11185
`\textregistered` 1249, 1250, 1829, 11054
`\textrightarrow` ... 1741, 11086, 11087
`\textrm` v15
`\textrquill` 1816, 11178, 11179
`\textsc` v21
`\textsection` 1228, 1260, 1466, 1612, 1822, 11049, m76
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx, t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx, y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx, D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx, I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx, N=ltluatex.dtx, O=ltfinal.dtx

- `\textservicemark` .. 1814, 11174, 11175
`\textsevenoldstyle` . 1755, 11106, 11107
`\textsf` v15
`\textsixoldstyle` .. 1754, 11104, 11105
`\textsl` v21
`\textsterling` 1216, 1268, 1369,
1467, 1686, 1818, 11046, 11195, 11197
`\textstyle` j15, t377, z63
`\textsubscript` G386, G397, G398
`\textsuperscript` 1252, 1254, 1255, G382
`\textsurd` 1842, 11186, 11187
`\TextSymbolUnavailable` 13, 1642
`\textthreeoldstyle` . 1751, 11098, 11099
`\textthreequarters` 1845, 11068
`\textthreequartersemdash` . 1739, 11023
`\textthreesuperior` 1834, 11059
`\texttilde` 1781, 11032
`\texttimes` 1847, 11069
`\texttrademark` 1252, 1806, 11044
`\texttt` v15
`\texttwelveudash` 1738, 11022
`\texttwooldstyle` .. 1750, 11096, 11097
`\texttwosuperior` 1833, 11058
`\textunderscore` 1239, 1264, 1468
`\textup` v21
`\textuparrow` 1772, 11120, 11121
`\textvisiblespace` 1241, 1469
`\textwidth` k18, B269,
G270, K86, K151, K175, K192,
K601, K611, K2112, K2144, O84
`\textwon` 1797, 11144, 11145
`\textyen` 1820, 11047
`\textzerooldstyle` . 1748, 11092, 11093
`\tf@size` . o188, o208, o482, p282, p284
`\TH` 1425, O328
`\th` 1470, O328
`\thanks` 344, F9
thebibliography (environment) .. 376
`\theequation` ... z245, z257, z316, z377
`\thefootnote` . G376, G430, G435, G455
`\thempfn` B271,
G406, G411, G446, G451, G454
`\thempfootnote` B271, G378
`\thepage`
k73, w6, x14, x34, F143, H15,
H32, J14, K218, K249, K1737, I23
`\Theta` t218
`\theta` t194
`\thicklines` D37
`\thickmuskip` t532, z146
`\thinlines` D37
`\thinmuskip` i282, t530, z144, z147
`\thinspace` i282, i303, z119, z148
`\thispagestyle` J6
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- `\thr@@` . b16, b481, p58, p208, p214,
p227, p234, p241, p246, z268,
z397, A232, A243, D144, D145,
D147, D148, D180, D202, O64, O72
`\tilde` t427
`\time` a174, a178
`\times` t310
`\title` 344, F3
`\to` t350
`\today` a179, a183, a191, a194, F8
`\toks` b31, b63, b88,
r453, r454, r464, r473, N36, O378
`\toks@` b41,
c59, c63, c66, c70, i264, i265,
i270, o113, o117, o119, o122,
o186, o191, r6, r7, r259, r263,
r269, r272, r277, r323, r324,
r326, r327, r357, r359, r363,
r380, r383, r442, r454, r455,
r456, r502, r504, r510, r518,
r522, r534, r537, r540, r548,
r550, r879, r881, r883, r886,
r888, r891, r894, r926, r927,
K2098, K2099, K2100, K2101,
L135, L136, L138, L139, L378, L379
`\toksdef` b46, b63, b88, N220
`\tokszero` N220
`\tolerance` b312, o497, J44, J52
`\top` t241
`\topfigrule` K637, K2214
`\topfraction` G277, K2180
`\topmargin` K78, K595
`\topmark` K2098, K2107
`\topsep` z382, A2, A59
`\topskip` b351, k49, A1, K135
`\totalheight` B32, B33, B34
`\tracefloats` K1807
`\tracefloatsoff` K1807
`\tracefloatvals` K1807
`\tracingall` b454
`\tracingassigns` b483, b516
`\tracingcommands`
... b465, b481, b490, b502, b519
`\tracingfonts` p17, p54, p58,
p86, p116, p125, p148, p178,
p192, p208, p214, p227, p234,
p241, p246, p255, p268, p276, p279
`\tracinggroups` b474, b527
`\tracingifs` b475, b526
`\tracinglostchars`
b328, b460, b472, b491, b510, b530
`\tracingmacros`
... b464, b480, b492, b509, b529
`\tracingnesting` b477, b524

- `\tracingnone` [b496](#)
- `\tracingoff` [p116](#), [p276](#)
- `\tracingon` [p117](#), [p277](#)
- `\tracingonline` [b449](#), [b501](#), [b518](#), [b539](#)
- `\tracingoutput` [b450](#), [b505](#), [b522](#), [b536](#)
- `\tracingpages`
... [b459](#), [b471](#), [b491](#), [b511](#), [b531](#)
- `\tracingparagraphs`
... [b461](#), [b473](#), [b492](#), [b508](#), [b528](#)
- `\tracingrestores`
... [b466](#), [b482](#), [b492](#), [b507](#), [b517](#)
- `\tracingscantokens`
..... [b456](#), [b476](#), [b499](#), [b525](#)
- `\tracingstats`
... [b458](#), [b470](#), [b490](#), [b512](#), [b532](#), [O2](#)
- `\triangle` [t247](#)
- `\triangleleft` [t276](#), [t390](#)
- `\triangleright` [t277](#), [t390](#)
- `\trivlist` [y73](#), [y80](#), [y86](#),
[y100](#), [z367](#), [A89](#), [C67](#), [E35](#), [E37](#)
- `\try@load@fontshape`
... [o306](#), [o314](#), [o386](#), [p474](#), [r208](#), [r225](#)
- `\try@simple@size` [p310](#), [p435](#)
- `\try@simples` [p393](#), [p399](#), [p403](#)
- `\try@size@range` [p101](#), [p310](#), [p386](#)
- `\try@size@substitution` .. [p103](#), [p390](#)
- `\tryif@simple` [p401](#), [p402](#)
- `\tryis@simple` [p402](#)
- `\ttdefault` [s12](#), [t29](#)
- `\ttfamily` [s10](#), [s11](#), [v17](#), [y120](#)
- `\tw@` [b16](#)
- `\two@digits` [a81](#),
[a180](#), [a181](#), [d2](#), [p466](#), [L431](#), [L462](#)
- `\twocolumn` [K173](#)
- `\type@restoreinfo` [p156](#), [p161](#)
- `\typein` [36](#), [36](#), [d18](#)
- `\typeout` [36](#), [a68](#), [a111](#), [a167](#),
[a192](#), [a194](#), [a206](#), [a221](#), [a228](#),
[a239](#), [a252](#), [a265](#), [a278](#), [a292](#),
[c20](#), [c37](#), [c42](#), [c47](#), [d3](#), [d23](#),
[d30](#), [g74](#), [k65](#), [k172](#), [k173](#), [k175](#),
[k210](#), [k220](#), [k223](#), [o300](#), [s120](#),
[s127](#), [s138](#), [s149](#), [s159](#), [t9](#), [t44](#),
[H8](#), [H25](#), [K1808](#), [L102](#), [O207](#),
[O332](#), [O339](#), [O351](#), [O352](#), [O360](#)

U

- `\u` [l187](#), [l313](#),
[l394](#), [l473](#), [l480](#), [l500](#), [l507](#), [l638](#)
- `\uccode` [O167](#), [O177](#),
[O186](#), [O188](#), [O192](#), [O194](#), [O283](#),
[O293](#), [O302](#), [O304](#), [O308](#), [O310](#)
- `\uchyph` [b329](#)
- File Key:** [a=ltmdirchk.dtx](#), [b=ltplain.dtx](#), [c=ltvers.dtx](#), [d=ltdefs.dtx](#),
[e=ltalloc.dtx](#), [f=ltcntrl.dtx](#), [g=lterror.dtx](#), [h=ltpar.dtx](#), [i=ltspace.dtx](#),
[j=ltlogos.dtx](#), [k=ltfiles.dtx](#), [l=ltoutenc.dtx](#), [m=ltcounts.dtx](#), [n=ltlength.dtx](#),
[o=ltfssbas.dtx](#), [p=ltfssstrc.dtx](#), [q=ltfsscmp.dtx](#), [r=ltfssdcl.dtx](#), [s=ltfssini.dtx](#),
[t=fontdef.dtx](#), [u=preload.dtx](#), [v=ltfntcmd.dtx](#), [w=ltpageno.dtx](#), [x=ltxref.dtx](#),
[y=ltmiscen.dtx](#), [z=ltmath.dtx](#), [A=ltlists.dtx](#), [B=ltboxes.dtx](#), [C=lttab.dtx](#),
[D=ltpictur.dtx](#), [E=ltthm.dtx](#), [F=ltsect.dtx](#), [G=ltfloat.dtx](#), [H=ltidxglo.dtx](#),
[I=ltbibl.dtx](#), [J=ltpage.dtx](#), [K=ltoutput.dtx](#), [L=ltclass.dtx](#), [M=lthyphen.dtx](#),
[N=ltluatex.dtx](#), [O=ltfinal.dtx](#)
- `\Umathchar` [N30](#)
- `\Umathcode` [b120](#), [O87](#), [O277](#)
- `\unboldmath` [s65](#)
- `\UndeclareTextCommand` [l141](#), [l1195](#),
[l1196](#), [l1197](#), [l1198](#), [l1199](#), [l1200](#)
- `\undefined` [a9](#),
[a11](#), [a17](#), [O95](#), [O107](#), [O108](#), [O128](#)
- `\undefinedpagestyle` [J4](#), [J8](#)
- `\underbar` [b419](#)
- `\underbrace` [t447](#)
- `\underline` [285](#), [b419](#), [B325](#), [B326](#)
- `\unhcopy` [b421](#), [C304](#), [D417](#)
- `\unicodedataline`
.. [N146](#), [N149](#), [N163](#), [N164](#), [N165](#)
- `\unicoderead` [N146](#),
[N160](#), [N161](#), [N162](#), [N163](#), [N168](#)
- `\uninstall` [N722](#)
- `\unitlength` [B48](#), [B58](#), [D5](#), [D12](#), [D13](#),
[D14](#), [D15](#), [D22](#), [D23](#), [D26](#), [D27](#),
[D34](#), [D57](#), [D115](#), [D168](#), [D170](#),
[D183](#), [D188](#), [D190](#), [D205](#), [D207](#),
[D210](#), [D250](#), [D251](#), [D295](#), [D296](#),
[D330](#), [D341](#), [D371](#), [D372](#), [D374](#),
[D375](#), [D378](#), [D379](#), [D381](#), [D382](#),
[D391](#), [D393](#), [D395](#), [D397](#), [O76](#)
- `\unless` [N154](#), [N162](#), [N164](#)
- `\unlhd` [s112](#)
- `\unpenalty` [v99](#), [y116](#)
- `\unrestored@protected@xdef`
[d230](#), [G411](#), [G435](#), [G451](#), [J21](#), [J35](#)
- `\unrhd` [s114](#)
- `\unsetattribute` [474](#), [N85](#), [N232](#)
- `\unvcopy` [z123](#)
- `\Uparrow` [t488](#)
- `\uparrow` [t482](#)
- `\upbracefill` [t450](#), [t466](#)
- `\update@ucllc@with@cyrillic`
..... [l851](#), [l879](#), [l903](#), [l909](#)
- `\updefault` [s21](#), [t34](#), [t41](#)
- `\Updownarrow` [t492](#)
- `\updownarrow` [t486](#)
- `\uplus` [t292](#)
- `\upper@bound` .. [p337](#), [p338](#), [p339](#), [p352](#)
- `\uppercase` [O318](#)
- `\upshape` [l366](#), [l616](#), [l683](#),
[l765](#), [s19](#), [s20](#), [s37](#), [s43](#), [s91](#), [v24](#)
- `\Upsilon` [t223](#)
- `\upsilon` [t205](#)
- `\use@mathgroup`
..... [o450](#), [o468](#), [o470](#), [p253](#),
[r63](#), [r92](#), [r424](#), [r526](#), [r529](#), [r904](#), [r928](#)
- `\usebox` [B109](#)
- `\usecounter` [A225](#), [A238](#)
- `\usefont` [o45](#), [o247](#), [s81](#), [s94](#)

- `\usepackage` L209, L239
`\UseTextAccent` 1110, 1138,
11004, 11189, 11190, 11192, 11193
`\UseTextSymbol` 1110, 1136, 11003, 11072
- V**
- `\v` 1188, 1314, 1393, 1476,
1477, 1478, 1482, 1484, 1487, 1489,
1491, 1497, 1503, 1504, 1505, 1509,
1511, 1514, 1516, 1518, 1524, 1639
`\v@false` z78
`\v@true` z77, z79
`\vadjust` i10,
i38, i47, i233, i249, G201, G223
`\valign` 1995
`\value` 132, m14, 19
`\varbigtriangledown` t280
`\varbigtriangleup` t281
`\varepsilon` j15, t210
`\varphi` t215
`\varpi` t212
`\varrho` t213
`\varsigma` t214
`\vartheta` t211
`\vbadness` b314, K2095
`\vdash` t318
`\vdots` t418
`\vec` t432
`\vector` g223, D112
`\vee` t284, t285
`\verb` y130, y132, y141
`\verb@balance@group`
y124, y125, y140, y142
`\verb@egroup` .. y125, y129, y140, y143
`\verb@eol@error` y126, y134
`\verbatim` y118
`verbatim*` (environment) y121
`\verbatim@font` y114, y120, y135
`\verbatim@nolig@list` y145, y151
`\version@elt` r18, r31, r32, r256, r257,
r306, r326, r417, r455, r547, r884
`\version@list` r16,
r21, r32, r249, r257, r311, r332,
r351, r422, r467, r497, r552, r897
`\Vert` t477, t479
`\vert` t480
`\vfil` b408, 1996, 1999, D274,
D317, K170, K386, K433, K598
`\vfilneg` b408
`\vfuzz` b337, J47, J54
`\vgl@` b398, b399
`\vglue` b398
`\vline` C325
File Key: a=ltmdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefs.dtx,
e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx,
j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx,
o=ltfssbas.dtx, p=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx
- `\voidb@x` b290, b429, n18
`\vphantom` z75, z94
`\vrule` b402, i298,
l242, l244, p144, t464, t465,
t467, t468, B118, B120, B170,
B177, B324, B367, C159, C192,
C306, C325, D106, D156, D159,
D175, D182, D197, D204, D273,
D317, D401, K1761, K2115, K2148
`\vspace` i226, i256, i257, i258
`\vsplit` K356, K403, K2097
- W**
- `\warn@rel@i` q5, q25,
q29, q81, q85, q90, q95, q119, q140
`\wedge` t282, t283
`\whatsit` N188
`\widehat` t435
`\widetilde` t434
`\widowpenalties` b99
`\widowpenalty` b321
`\width` B29
`\wlog` a95, b40,
b138, b223, b236, b265, b280,
L104, N6, N7, N8, N54, O40, O385
`\wp` t232
`\wr` t296
`\wrong@fontshape` o310, o368
- X**
- `\x` o267, o268
`\x@protect` d208, d219, d267
`\xe@alloc@` O36, O46
`\xe@alloc@intercharclass` O21
`\xe@ch@ck` O37, O41
`\XeTeXcharclass` O25,
O28, O34, O47, O53, O62, O69
`\XeTeXcharclassCL` O100
`\XeTeXcharclassCM` O104
`\XeTeXcharclassEX` O101
`\XeTeXcharclassID` O98
`\XeTeXcharclassIS` O102
`\XeTeXcharclassNS` O103
`\XeTeXcharclassOP` O99
`\XeTeXdashbreakstate` O204
`\XeTeXintercharclasses` .. O94, O127
`\XeTeXinterchartoks` .. O95, O109,
O110, O111, O112, O113, O114,
O115, O116, O117, O118, O119,
O120, O121, O122, O123, O128,
O133, O134, O135, O136, O137,
O138, O139, O140, O141, O142,
O143, O144, O145, O146, O147
`\XeTeXmathcode` O88, O278

<code>\XeTeXuseglyphmetrics</code> ..	O201, O203	<code>\yxdim</code>	D353
<code>\Xi</code>	t220		
<code>\xi</code>	t200		
<code>\xtxHanGlue</code>	O107, O131, O139, O140, O141, O142, O143, O144, O145, O146, O147		
<code>\xtxHanSpace</code> ..	O108, O132, O133, O134, O135, O136, O137, O138		
	Y		
<code>\year</code>	a180, c13, L462		
		Z	
		<code>\Z</code>	O181, O265, O297
		<code>\z</code>	O170, O266, O286
		<code>\z@</code>	b290
		<code>\zap@skip</code>	b290
		<code>\zap@space</code>	k84, L124, L272 , L289, L306
		<code>\zeta</code>	t192

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=ltfssstrc.dtx, q=ltfsscmp.dtx, r=ltfssdcl.dtx, s=ltfssini.dtx,
t=fontdef.dtx, u=preload.dtx, v=ltfntcmd.dtx, w=ltpageno.dtx, x=ltxref.dtx,
y=ltmiscen.dtx, z=ltmath.dtx, A=ltlists.dtx, B=ltboxes.dtx, C=lttab.dtx,
D=ltpictur.dtx, E=ltthm.dtx, F=ltsect.dtx, G=ltfloat.dtx, H=ltidxglo.dtx,
I=ltbibl.dtx, J=ltpage.dtx, K=ltoutput.dtx, L=ltclass.dtx, M=lthyphen.dtx,
N=ltluatex.dtx, O=ltfinal.dtx