The LaTeX 2_ε Sources

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

2011/06/27

Contents

a ltdirchk.dtx 1

1 LaTeX System Dependent Initialisations 1

2 Initialisation 2
- 2.1 INITEX 2
- 2.2 Some bits of 2e 3

3 texsys.cfg 3
- 3.1 texsys.cfg 4
- 3.2 UNIX (web2c) 5
- 3.3 UNIX (other) 5
- 3.4 MSDOS (emtex) 5
- 3.5 MSDOS (other) 5
- 3.6 VMS (DECUS TeX, PD VMS 3.6) 5
- 3.7 VMS (???) 6
- 3.8 MACINTOSH (OzTeX 1.6) 6
- 3.9 MACINTOSH (other) 6
- 3.10 FAKE EXAMPLE 6

4 Setting `\@currdir` 7

5 Setting `\input@path` 8

6 Filename Parsing 9

7 TeX Versions 10

8 ltxcheck.tex 10

b ltplain.dtx 11

9 Plain TeX 11

c ltvers.dtx 21

10 Version Identification 21

d ltdefns.dtx 22

11 Definitions 22
- 11.1 Initex initialisations . . . 22
- 11.2 Saved versions of TEX primitives . . . 22
- 11.3 Command definitions . . . 23
- 11.4 Robust commands and protect . . . 29
- 11.5 Internal defining commands . . . 32
- 11.6 Commands for Autoloading . . . 33

e ltalloc.dtx 34

12 Counters 34

f ltcntrl.dtx 35

13 Program control structure 35

g lterror.dtx 38

14 Error handling 38
- 14.1 General commands . . . 38
- 14.2 Specific errors . . . 42

h ltpar.dtx 46

15 Paragraphs 46
- 15.1 Implementation . . . 46

i ltspace.dtx 48

16 Spacing 48
- 16.1 User Commands . . . 48
- 16.2 Chris' comments . . . 48
- 16.3 Some immediate actions . . . 50
- 16.4 The code . . . 50
- 16.5 Vertical spacing . . . 54
- 16.6 Horizontal space (and breaks) . . . 56

j ltlogos.dtx 59

17 Logos 59

k ltfiles.dtx 60

18 File Handling 60
- 18.1 Safe Input Macros . . . 65
- 18.2 Listing files . . . 67

l ltoutenc.dtx 69

19 Font encodings 69
19.1 Removing encoding-specific commands 71
19.2 The order of declarations 71
19.3 Docstrip modules 72
19.4 Definitions for the kernel 72
19.4.1 Declaration commands 72
19.4.2 Hyphenation 78
19.4.3 Miscellania 78
19.4.4 Default encodings 78
19.4.5 Math material 80
19.5 Definitions for the OT1 encoding 81
19.6 Definitions for the T1 encoding 82
19.7 Definitions for the OMS encoding 86
19.8 Definitions for the OML encoding 87
19.9 Definitions for the OT4 encoding 87
19.10Definitions for the TS1 encoding 89

20 Package files 92
20.1 The fontenc package 93
20.2 The textcomp package 94
20.2.1 Supporting oldstyle digits 100
20.2.2 Subset encoding defaults 101

m ltcounts.dtx 103

21 Counters and Lengths 103
21.1 Environment Counter Macros 103

n ltlength.dtx 106

22 Lengths 106

o ltfssbas.dtx 107

23 Autoloading parts of NFSS 107

24 Preliminary macros 107

25 Macros for setting up the tables 108

26 Selecting a new font 112
26.1 Macros for the user 112
26.2 Macros for loading fonts 115

27 Assigning math fonts to *versions* 120

p ltfsstrc.dtx 125

28 Introduction 125

29 A driver for this document 125

30 The Implementation 125

31 Handling Options **126**

32 Macros common to `fam.tex` and `tracefnt.sty` **127**
32.1 General font loading . . . 127
32.2 Math fonts setup . . . 131
32.2.1 Outline of algorithm for math font sizes . . . 131
32.2.2 Code for math font size setting . . . 132
32.2.3 Other code for math . . . 133

33 Scaled font extraction **135**
33.1 Sizefunctions . . . 141

q ltfsscmp.dtx **145**

34 Compatibility code for NFSS release 1 **145**

r ltfssdcl.dtx **149**

35 Interface Commands **149**

s ltfssini.dtx **168**

36 NFSS Initialisation **168**
36.1 Providing math *versions* . . . 168
36.2 Miscellaneous . . . 169

t fontdef.dtx **173**

37 Introduction **173**

38 Customization **173**

39 The `docstrip` modules **174**

40 A driver for this document **174**

41 The `fonttext.ltx` file **174**
41.1 Encodings . . . 174
41.2 Defaults . . . 176

42 The `fontmath.ltx` file **176**
42.1 The font encodings used . . . 176
42.1.1 Symbolfont and Alphabet declarations . . . 177
42.2 Math font sizes . . . 177
42.3 The math symbol assignments . . . 178
42.3.1 The letters . . . 178
42.3.2 The digits . . . 179
42.3.3 Punctuation, brace, etc. keys . . . 179
42.3.4 Delimitercodes for characters . . . 179
42.4 Symbols accessed via control sequences . . . 180
42.4.1 Greek letters . . . 180
42.4.2 Ordinary symbols . . . 181
42.4.3 Large Operators . . . 181
42.4.4 Binary symbols . . . 181
42.4.5 Relations . . . 182
42.4.6 Arrows . . . 183

42.4.7 Punctuation symbols 184
42.4.8 Math accents 184
42.4.9 Radicals 185
42.4.10 Over and under something, etc 185
42.4.11 Delimiters 185
42.5 Math versions of text commands 186
42.6 Other special functions and parameters 186
42.6.1 Bigggggg 186
42.6.2 The log-like functions 187
42.6.3 Parameters 187

43 Default cfg files **187**

u preload.dtx **188**

44 Overview **188**

45 Customization **188**

46 Module switches for the DOCSTRIP program **188**

47 A driver for this document **189**

48 The code **189**

v ltfntcmd.dtx **191**

49 Introduction **191**

50 The implementation **193**

51 Initialization **197**

w ltpageno.dtx **198**

52 Page Numbering **198**

x ltxref.dtx **199**

53 Cross Referencing **199**
53.1 Cross Referencing 199
53.2 An extension of counter referencing 201

y ltmiscen.dtx **202**

54 Miscellaneous Environments **202**
54.1 Environments 202
54.2 Center, Flushright, Flushleft 205
54.3 Verbatim 207

z ltmath.dtx **210**

55 Math setup 210
55.1 Math commands based on plain TEX 210
55.1.1 The log-like functions 210
55.1.2 Bigggggg 211
55.1.3 The UNSORTED Rest 211
55.2 Math Environments 214
55.3 External options to the standard document classes 216
55.3.1 Left equation numbering 216
55.3.2 Flush left equations 217

A ltlists.dtx 219

56 List, and related environments 219
56.1 List and Trivlist 220
56.2 Vertical Spacing (skips) 220
56.3 Penalties 221
56.4 Horizontal Spacing (dimens) 221
56.5 Default Values 221
56.6 Itemize and Enumerate 229

B ltboxes.dtx 232

57 LATEX Box commands 232
57.1 Some low-level constructs 240

C lttab.dtx 241

58 Tabbing, Tabular and Array Environments 241
58.1 tabbing 241
58.2 array and tabular environments 248

D ltpictur.dtx 260

59 Picture Mode 260
59.1 Curves 276

E ltthm.dtx 279

60 Theorem Environments 279

F ltsect.dtx 282

61 Sectioning Commands 282
61.1 The Title 282
61.2 Sectioning 283
61.2.1 Initializations 288
61.3 Table of Contents etc. 288
61.3.1 Convention 288
61.3.2 Commands 288

G ltfloat.dtx 291

62 Floats **291**
62.1 Floating Environments 291
62.2 Footnotes 300

H ltidxglo.dtx **305**

63 Index and Glossary Generation **305**

I ltbibl.dtx **307**

64 Bibliography Generation **307**
64.1 Default definitions 309

J ltpage.dtx **311**

65 Page styles and related commands **311**
65.1 Page Style Commands 311
65.2 How a page style makes running heads and feet 311
65.3 marking conventions 311

K ltoutput.dtx **314**

66 Output Routine **314**
66.1 Floats 314
66.1.1 Kludgeins 349
66.1.2 Float control 351
66.1.3 Float placement parameters 359

L ltclass.dtx **362**

67 Introduction **362**

68 User interface **362**
68.1 Option processing 363

69 Class and Package interface **363**
69.1 Class name and version 363
69.2 Package name and version 363
69.3 Requiring other packages 363
69.4 Declaring new options 364
69.5 Safe Input Macros 365

70 Implementation **365**
70.1 Hooks 374
70.2 Providing shipment 375

71 After Preamble **377**

M lthyphen.dtx **378**

N ltfinal.dtx **379**

72 Final settings **379**
72.1 Debugging 379
72.2 Typesetting parameters 379
72.3 Lccodes for hyphenation 379
72.4 Hyphenation 380
72.5 Font loading 381
72.6 Input encoding 381
72.7 Lccodes and uccodes 382
72.8 Applying Patch files 383
72.9 Freeing Memory 384
72.10Initialise file list 385
72.11Dumping the format 385

O ltpatch **386**

Change History **387**

Index **432**

File a

ltdirchk.dtx

1 LaTeX 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, 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, `/`.

`\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 TeX 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. LaTeX 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 LaTeX to take appropriate action for these old TeXs.

If your TeX is older than version 3.141, then you should define `\@TeXversion` (using `\def`) to be the version number. If you do not do this[1] , LaTeX will not work around a bug in old TeX 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.
 ...

l.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.
! .
 ...

l.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 TeX, 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

```
⟨*dircheck⟩
⟨*initex⟩
⟨initex⟩\ifnum\catcode`\{=1
⟨initex⟩  \errmessage
⟨initex⟩  {LaTeX must be made using an initex with no format preloaded}
⟨initex⟩\fi
\catcode`\{=1
\catcode`\}=2
\catcode`\#=6
\catcode`\^=7
\chardef\active=13
\catcode`\@=11
\countdef\count@=255
\let\bgroup={ \let\egroup=}
\ifx\@@input\@undefined\let\@@input\input\fi
\ifx\@@end\@undefined\let\@@end\end\fi
\chardef\@inputcheck0
\chardef\sixt@@n=16
\newlinechar`\^^J
\def\typeout{\immediate\write17}
\def\dospecials{\do\ \do\\\do\{\do\}\do\$\do\&%
  \do\#\do\^\do\_\do\%\do\~}
\def\@makeother#1{\catcode`#1=12\relax}
```

---

[1] Actually if your TeX is really old, version 2, LaTeX can detect this, and sets `\@TeXversion` to 2 if it is not set in the `cfg` file.

```
\def\space{ }
\def\@tempswafalse{\let\if@tempswa\iffalse}
\def\@tempswatrue{\let\if@tempswa\iftrue}
\let\if@tempswa\iffalse
\def\loop#1\repeat{\def\iterate{#1\relax\expandafter\iterate\fi}%
  \iterate \let\iterate\relax}
\let\repeat\fi
⟨/initex⟩
```

2.2 Some bits of 2e

```
⟨*2ekernel⟩
\def\two@digits#1{\ifnum#1<10 0\fi\number#1}
\long\def\@firstoftwo#1#2{#1}
\long\def\@secondoftwo#1#2{#2}
```

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

```
\def\ProvidesFile#1{%
  \begingroup
    \catcode`\ 10 %
    \ifnum \endlinechar<256 %
      \ifnum \endlinechar>\m@ne
        \catcode\endlinechar 10 %
      \fi
    \fi
    \@makeother\/%
    \@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}}
\def\@providesfile#1[#2]{%
    \wlog{File: #1 #2}%
    \@addtofilelist{ #2}%
    \endgroup}
\long\def\@addtofilelist#1{}
```

```
\def\@empty{}
\catcode`\%=12
\def\@percentchar{%}
\catcode`\%=14
\let\@currdir\@undefined
\let\input@path\@undefined
\let\filename@parse\@undefined
```

`\strip@prefix`

```
\def\strip@prefix#1>{}
⟨/2ekernel⟩
```

3 texsys.cfg

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

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

```
⟨*docstrip⟩
\openin15=texsys.cfg-not-found
\ifeof15
\typeout{** Writing a default texsys.cfg}
\immediate\openout15=texsys.cfg
\begingroup
\catcode`\^^M\active%
```

```
\let^^M\par%
\def\reserved@a#1^^M{%
 \def\reserved@b{#1}%
 \ifx\reserved@b\reserved@c\endgroup\else%
     \immediate\write15{#1}%
     \expandafter\reserved@a\fi}%
\def\reserved@d#1START^^M{\let\do\@makeother\dospecials\reserved@a}%
\catcode`\%=12
\def\reserved@c{%END}
\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, TEXwill try to load the expansion of

⟨*dir*⟩⟨*filename*⟩⟨*space*⟩

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

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

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

Normally one would not need to define this macro in `texsys.cfg` as the automatic tests can supply parsers that work with UNIX and VMS syntax, as well as a basic parser that willcover 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.

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

```
% \def\@currdir{./}
% \def\input@path{%
%   {/usr/local/lib/tex/inputs/distrib/}%
%   {/usr/local/lib/tex/inputs/contrib/}%
%   {/usr/local/lib/tex/inputs/local/}%
% }
```

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.

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

```
% \def\@currdir{./}
% \def\input@path{%
%   {c:/tex/inputs/distrib/}%
%   {c:/tex/inputs/contrib/}%
%   {c:/tex/inputs/local/}%
% }
```

3.6 VMS (DECUS TeX, 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.

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

```
% \def\@currdir{[]}
% \def\input@path{%
%   {tex_inputs:}%
%   {SOMEDISK:[SOME.TEX.DIRECTORY]}%
% }
```

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.

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

```
% \def\@currdir{:}
% \def\input@path{%
%   {Hard-Disk:Applications:TeX:TeX-inputs:}%
%   {Hard-Disk:Applications:TeX:My-inputs:}%
% }
```

3.10 FAKE EXAMPLE

This example is for an operating system that has filenames of the form `<area>name` For maximum compatability 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:

```
% \def\@dir#1#2 {%
%   \@d@r{#1}#2..\@nil}
% \def\@d@r#1#2.#3.#4\@nil{%
%   <\ifx\@dir#1\@dir\else#1\ifx\@dir#3\@dir\else.\fi\fi#3>#2 }
%
% \def\@currdir{\@dir{}}
% \def\input@path{%
%   {\@dir{area.one}}%
%   {\@dir{area.two}}%
% }
```

END

```
\immediate\closeout15
```

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

```
\else
\typeout{** Using the existing texsys.cfg}
\closein15
\input texsys.cfg
\fi
⟨/docstrip⟩
```

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

```
⟨dircheck⟩\input texsys.cfg
```

4 Setting `\@currdir`

`\@currdir`
`\IfFileExists`

This is a local definition of `\IfFileExists`. It tries to relocate `texsxys.aux`. If 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.

```
\begingroup
\count@\time
\divide\count@ 60
\count2=-\count@
\multiply\count2 60
\advance\count2 \time
```

`\today`

The current date and time stamp.

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

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

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

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

```
\def\IfFileExists#1#2#3{%
  \openin\@inputcheck#1 %
  \ifeof\@inputcheck
     #3\relax
  \else
    \read\@inputcheck to \reserved@a
    \ifx\reserved@a\today
      \typeout{#1 found}#2\relax
    \else
      \typeout{BAD: old file \reserved@a (should be \today)}%
      #3\relax
    \fi
  \fi
  \closein\@inputcheck}
```

```
\endlinechar=-1
```

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

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

```
  \ifx\@currdir\@undefined
    \global\let\@currdir\@empty
    \typeout{^^J^^J%
```

```
      !! No syntax for the current directory could be found^^J%
      }%
  \fi
```

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

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

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

```
\fi

\immediate\closeout15 %
\endgroup

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

Stop here if the file is being used unstripped.

```
⟨*docstrip⟩
\relax\endinput
⟨/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 LaTeX 2ε 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 LaTeX 2ε. 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.

```
  \typeout{^^J%
    Assuming \noexpand\openin and \noexpand\input^^J%
    \ifx\input@path\@undefined
```

`\input@path` has not been pre-defined.

```
      have the same search path.^^J%
    \else
```

`\input@path` has been defined in `texsys.cfg`.

```
      have different  search paths.^^J%
      LaTeX will use the path specified by \noexpand\input@path:^^J%
    \fi
    }
```

6 Filename Parsing

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

```
\ifx\filename@parse\@undefined
  \def\reserved@a{./}\ifx\@currdir\reserved@a
```

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

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

Search for the last `/`.

```
    \def\filename@path#1/#2\\{%
      \ifx\\#2\\%
         \def\reserved@a{\filename@simple#1.\\}%
      \else
         \edef\filename@area{\filename@area#1/}%
         \def\reserved@a{\filename@path#2\\}%
      \fi
      \reserved@a}
```

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

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

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

Search for the last `]`.

```
    \def\filename@path#1]#2\\{%
      \ifx\\#2\\%
         \def\reserved@a{\filename@simple#1.\\}%
      \else
         \edef\filename@area{\filename@area#1]}%
         \def\reserved@a{\filename@path#2\\}%
      \fi
      \reserved@a}
```

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

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

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

Search for the last `:`.

```
    \def\filename@path#1:#2\\{%
      \ifx\\#2\\%
         \def\reserved@a{\filename@simple#1.\\}%
      \else
         \edef\filename@area{\filename@area#1:}%
         \def\reserved@a{\filename@path#2\\}%
      \fi
      \reserved@a}
```

```
  \else
```

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

```
    \typeout{^^JDefining generic filename parser.^^J}
    \def\filename@parse#1{%
      \let\filename@area\@empty
      \expandafter\filename@simple#1.\\}
  \fi\fi\fi
```

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

```
  \def\filename@simple#1.#2\\{%
    \ifx\\#2\\%
       \let\filename@ext\relax
    \else
       \edef\filename@ext{\filename@dot#2\\}%
    \fi
    \edef\filename@base{#1}}
```

Remove a final dot, added earlier.

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

```
\else
```

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

```
  \typeout{^^J^^J%
    \noexpand\filename@parse was defined in texsys.cfg:^^J%
    \expandafter\strip@prefix\meaning\filename@parse.^^J%
    }
\fi
```

7 TEX Versions

`\@TeXversion`

TEX 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 TEX3.14 would be detected as TEX3, but LATEX only needs to take account of TEX's older than 3, or between 3 and 3.14.

```
\ifx\@TeXversion\@undefined
  \ifx\@undefined\inputlineno
    \def\@TeXversion{2}
  \else
   {\catcode`\^^J=\active
     \def\reserved@a#1#2\@@{\if#1\string^3\fi}
     \edef\reserved@a{\expandafter\reserved@a\string^^J\@@}
     \ifx\reserved@a\@empty\else\gdef\@TeXversion{3}\fi}
  \fi
\fi
```

```
⟨/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

ltplain.dtx

9 Plain TeX

LaTeX includes almost all of the functionality of Knuth's original 'Basic Macros' That is, the plain TeX format described in Appendix B of the TeXBook. 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.

LaTeX font definitions are done using NFSS2 so none of PLAIN's font definitions are in LaTeX.

LaTeX has its own tabbing environment, so PLAIN's is disabled.

LaTeX uses its own output routine, so most of the plain one was removed.

```
⟨*2ekernel | autoload⟩
\catcode`\{=1 % left brace is begin-group character
\catcode`\}=2 % right brace is end-group character
\catcode`\$=3 % dollar sign is math shift
\catcode`\&=4 % ampersand is alignment tab
\catcode`\#=6 % hash mark is macro parameter character
\catcode`\^=7 % circumflex and uparrow are for superscripts
\catcode`\_=8 % underline and downarrow are for subscripts
\catcode`\^^I=10 % ascii tab is a blank space
\chardef\active=13 \catcode`\~=\active % tilde is active
\catcode`\^^L=\active \outer\def^^L{\par}% ascii form-feed is \outer\par
```

```
\message{catcodes,}
```

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

```
\def\dospecials{\do\ \do\\\do\{\do\}\do\$\do\&%
  \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.

```
\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` `\tw@` `\thr@@` `\sixt@@n` `\@cclv`

Small constants are defined using `\chardef`.

```
\chardef\@ne=1
\chardef\tw@=2
\chardef\thr@@=3
\chardef\sixt@@n=16
\chardef\@cclv=255
```

`\@cclvi` `\@m` `\@M` `\@MM`

Constants above 255 defined using `\mathchardef`.

```
\mathchardef\@cclvi=256
\mathchardef\@m=1000
\mathchardef\@M=10000
\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.

```
\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, `\box`255 is reserved for `\output`; `\count`255, `\dimen`255, and `\skip`255 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.

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

`\insc@unt`
`\allocationnumber`

The insertion counter and most recent allocation.

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

`\m@ne`

The constant -1.

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

`\wlog`

Write on log file (only)

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

`\count@`
`\dimen@`
`\dimen@i`
`\dimen@ii`
`\skip@`
`\toks@`

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

```
\countdef\count@=255
\dimendef\dimen@=0
\dimendef\dimen@i=1 % global only
\dimendef\dimen@ii=2
\skipdef\skip@=0
\toksdef\toks@=0
```

`\newcount`
`\newdimen`
`\newskip`
`\newmuskip`
`\newbox`
`\newhelp`
`\newtoks`

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

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

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

LaTeX 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.

```
\def\newcount{\alloc@0\count\countdef\insc@unt}
\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}
\def\newskip{\alloc@2\skip\skipdef\insc@unt}
\def\newmuskip{\alloc@3\muskip\muskipdef\@cclvi}
\def\newbox{\alloc@4\box\chardef\insc@unt}
\def\newhelp#1#2{\newtoks#1#1\expandafter{\csname#2\endcsname}}
\def\newtoks{\alloc@5\toks\toksdef\@cclvi}
```

`\newread`
`\newwrite`

```
\def\newread{\alloc@6\read\chardef\sixt@@n}
\def\newwrite{\alloc@7\write\chardef\sixt@@n}
```

LaTeX defines `\newfam` in `ltfss.dtx`.

```
\def\newfam{\alloc@8\fam\chardef\sixt@@n}
```

`\newlanguage`

```
\def\newlanguage{\alloc@9\language\chardef\@cclvi}
```

\alloc@

```
\def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
  \ch@ck#1#4#2% make sure there’s still room
  \allocationnumber\count1#1%
  \global#3#5\allocationnumber
  \wlog{\string#5=\string#2\the\allocationnumber}}
```

\newinsert

```
\def\newinsert#1{\global\advance\insc@unt \m@ne
  \ch@ck0\insc@unt\count
  \ch@ck1\insc@unt\dimen
  \ch@ck2\insc@unt\skip
  \ch@ck4\insc@unt\box
  \allocationnumber\insc@unt
  \global\chardef#1\allocationnumber
  \wlog{\string#1=\string\insert\the\allocationnumber}}
```

\ch@ck

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoload | autoerr⟩
\gdef\ch@ck#1#2#3{%
  \ifnum\count1#1<#2\else
⟨!autoload⟩    \errmessage{No room for a new #3}%
⟨autoload⟩      \@autoerr\ch@ck#1#2#3%
  \fi}
⟨/2ekernel | autoload | autoerr⟩
⟨*2ekernel | autoload⟩
```

\maxdimen
\hideskip

Here are some examples of allocation.

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

\p@
\z@
\z@skip
\voidb@x

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

```
\message{compatibility for TeX 2, }
```

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

```
\ifx\@undefined\inputlineno
  \newcount\inputlineno
```

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

```
  \inputlineno-1
```

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

```
  \ifx\@undefined\language
    \newcount\language
  \fi
  \newcount\lefthyphenmin
  \newcount\righthyphenmin
  \newcount\errorcontextlines
  \newcount\holdinginserts
  \newdimen\emergencystretch
  \newcount\badness
  \let\noboundary\relax
  \newcount\setlanguage
\fi
```

Assign initial values to TEX's parameters

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

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

```
\pretolerance=100
\tolerance=200 % INITEX sets this to 10000
\hbadness=1000
\vbadness=1000
\linepenalty=10
\hyphenpenalty=50
\exhyphenpenalty=50
\binoppenalty=700
\relpenalty=500
\clubpenalty=150
\widowpenalty=150
\displaywidowpenalty=50
\brokenpenalty=100
\predisplaypenalty=10000
```

```
\postdisplaypenalty=0
\interlinepenalty=0
\floatingpenalty=0, set during \insert
\outputpenalty=0, set before TeX enters \output
```

```
\doublehyphendemerits=10000
\finalhyphendemerits=5000
\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
```

```
\tracinglostchars=1
```

```
\tracingcommands=0
\tracingrestores=0
\language=0
```

```
\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
```

```
\defaulthyphenchar=`\-
\defaultskewchar=-1
```

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

```
\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.

```
\showboxbreadth=-1
\showboxdepth=-1
\errorcontextlines=-1
```

```
\hfuzz=0.1pt
\vfuzz=0.1pt
\overfullrule=5pt
\maxdepth=4pt
\splitmaxdepth=\maxdimen
\boxmaxdepth=\maxdimen
```

`\lineskiplimit`=0pt, changed by `\normalbaselines`

```
\delimitershortfall=5pt
\nulldelimiterspace=1.2pt
\scriptspace=0.5pt
```

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

```
\parindent=20pt
```

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

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

```
\parskip=0pt plus 1pt
\abovedisplayskip=12pt plus 3pt minus 9pt
\abovedisplayshortskip=0pt plus 3pt
\belowdisplayskip=12pt plus 3pt minus 9pt
\belowdisplayshortskip=7pt plus 3pt minus 4pt
```

`\leftskip`=0pt
`\rightskip`=0pt

```
\topskip=10pt
\splittopskip=10pt
```

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

```
\parfillskip=0pt plus 1fil
```

`\normalbaselineskip`
`\normallineskip`
`\normallineskiplimit`

We also define special registers that function like parameters:

```
\newskip\normalbaselineskip \normalbaselineskip=12pt
\newskip\normallineskip \normallineskip=1pt
\newdimen\normallineskiplimit \normallineskiplimit=0pt
```

`\interfootlinepenalty`

```
\newcount\interfootnotelinepenalty \interfootnotelinepenalty=100
```

Definitions for preloaded fonts

`\magstephalf`
`\magstep`

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

Macros for setting ordinary text

`\frenchspacing`
`\nonfrenchspacing`

```
\def\frenchspacing{\sfcode`\.\@m \sfcode`\?\@m \sfcode`\!\@m
  \sfcode`\:\@m \sfcode`\;\@m \sfcode`\,\@m}
\def\nonfrenchspacing{\sfcode`\.3000\sfcode`\?3000\sfcode`\!3000%
  \sfcode`\:2000\sfcode`\;1500\sfcode`\,1250 }
```

`\normalbaselines`

```
\def\normalbaselines{\lineskip\normallineskip
  \baselineskip\normalbaselineskip \lineskiplimit\normallineskiplimit}
```

`\M`
`\I`

Save a bit of space by using `\let` here.

```
\def\^^M{\ } % control <return> = control <space>
\let\^^I\^^M % same for <tab>
```

`\lq`
`\rq`

```
\def\lq{`}
\def\rq{'}
```

`\lbrack`
`\rbrack`

```
\def\lbrack{[}
\def\rbrack{]}
```

`\aa`
`\AA`

These are not from plain.tex but they are similar to other commands found here and nowhere else, being alternate input forms for characters.

```
\def \aa {\r a}
\def \AA {\r A}
```

`\endgraf`
`\endline`

```
\let\endgraf=\par
\let\endline=\cr
```

`\space`

```
\def\space{ }
```

`\empty`

This probably ought to go altogether, but let it to the LaTeX version to save space.

```
\let\empty\@empty
```

`\null`

```
\def\null{\hbox{}}
```

`\bgroup`
`\egroup`

```
\let\bgroup={
\let\egroup=}
```

`\obeylines`
`\obeyspaces`

In `\obeylines`, we say `\let^^M=\par` instead of `\def^^M{\par}` since this allows, for example, `\let\par=\cr \obeylines \halign{...`

```
{\catcode`\^^M=\active % these lines must end with %
  \gdef\obeylines{\catcode`\^^M\active \let^^M\par}%
  \global\let^^M\par} % this is in case ^^M appears in a \write
\def\obeyspaces{\catcode`\ \active}
{\obeyspaces\global\let =\space}
```

`\loop`
`\iterate`
`\repeat`

We use Kabelschacht's method of doing loops, see TUB 8#2 (1987). (unless that breaks something :-). It turned out to need an extra `\relax`: see pr/642 (`\loop` could do one iteration too much in certain cases).

```
\long\def \loop #1\repeat{%
  \def\iterate{#1\relax  % Extra \relax
               \expandafter\iterate\fi
               }%
  \iterate
  \let\iterate\relax
}
```

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

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

LaTeX defines `\smallskip`, etc. in `ltspace.dtx`.

`\nointerlineskip`
`\offinterlineskip`

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

`\vglue`
`\hglue`

```
\def\vglue{\afterassignment\vgl@\skip@=}
\def\vgl@{\par \dimen@\prevdepth \hrule \@height\z@
  \nobreak\vskip\skip@ \prevdepth\dimen@}
\def\hglue{\afterassignment\hgl@\skip@=}
\def\hgl@{\leavevmode \count@\spacefactor \vrule \@width\z@
  \nobreak\hskip\skip@ \spacefactor\count@}
```

LaTeX defines `~` in `ltdefns.dtx`.

`\slash`

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

`\break`
`\nobreak`
`\allowbreak`

```
\def\break{\penalty-\@M}
\def\nobreak{\penalty \@M}
\def\allowbreak{\penalty \z@}
```

`\filbreak`
`\goodbreak`

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

`\eject`

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

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

`\removelastskip`

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

`\smallbreak`
`\medbreak`
`\bigbreak`

```
\def\smallbreak{\par\ifdim\lastskip<\smallskipamount
  \removelastskip\penalty-50\smallskip\fi}
\def\medbreak{\par\ifdim\lastskip<\medskipamount
  \removelastskip\penalty-100\medskip\fi}
\def\bigbreak{\par\ifdim\lastskip<\bigskipamount
  \removelastskip\penalty-200\bigskip\fi}
```

`\m@th`

```
\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?).

```
\def\underbar#1{\underline{\sbox\tw@{#1}\dp\tw@\z@\box\tw@}}
```

`\strutbox` LaTeX sets `\strutbox` in `\set@fontsize`.

`\strut`

```
\newbox\strutbox
\def\strut{\relax\ifmmode\copy\strutbox\else\unhcopy\strutbox\fi}
```

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

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

`\narrower`

```
\def\narrower{%
  \advance\leftskip\parindent
  \advance\rightskip\parindent}
```

LaTeX defines `\ae` and similar commands elsewhere.

```
\chardef\%=`\%
\chardef\&=`\&
\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

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

`\mathhexbox`

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

`\ialign`

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

`\oalign`

`\o@lign`

`\ooalign`

```
\def\oalign#1{\leavevmode\vtop{\baselineskip\z@skip \lineskip.25ex%
  \ialign{##\crcr#1\crcr}}}
\def\o@lign{\lineskiplimit\z@ \oalign}
\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.

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

`\ltx@sh@ft` This is the LaTeX 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).

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

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

LaTeX change: Make `\t` work in a moving argument. Now defined elsewhere.

`\hrulefill`
`\dotfill`

LaTeX change: `\kern\z@` added to end of `\hrulefill` and `\dotfill` to make them work in 'tabular' and 'array' environments. (Change made 24 July 1987). LaTeX change: `\leavevmode` added at begining of `\dotfill` and `\hrulefill` so that they work as expected in vertical mode.

```
\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.

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

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

```
\sfcode`\)=0 \sfcode`\'=0 \sfcode`\]=0
```

The `\nonfrenchspacing` macro will make further changes to `\sfcode` values.

Definitions related to output

`\magnification` doesn't work in LaTeX.

```
\def\magnification{\afterassignment\m@g\count@}
\def\m@g{\mag\count@
  \hsize6.5truein\vsize8.9truein\dimen\footins8truein}
```

`\showoverfull`

The following commands are used in debugging:

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

`\showoutput`
`\loggingoutput`

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoerr⟩
\gdef\loggingoutput{\tracingoutput\@ne
    \showboxbreadth\maxdimen\showboxdepth\maxdimen\errorstopmode}
\gdef\showoutput{\loggingoutput\showoverfull}
⟨/2ekernel | autoerr⟩
⟨autoload⟩\def\showoutput{\@autoerr\showoutput}
```

`\tracingall`
`\loggingall`

```
⟨*2ekernel | autoerr⟩
\gdef\loggingall{\tracingcommands\tw@\tracingstats\tw@
  \tracingpages\@ne\tracinglostchars\@ne
  \tracingmacros\tw@\tracingparagraphs\@ne\tracingrestores\@ne
  \errorcontextlines\maxdimen\loggingoutput}
\gdef\tracingall{\loggingall\showoverfull}
⟨/2ekernel | autoerr⟩
⟨autoload⟩\def\tracingall{\@autoerr\tracingall}
```

LaTeX change: `\showhyphens` Defined later.

Punctuation affects the spacing.

```
⟨*2ekernel | autoload⟩
\nonfrenchspacing
⟨/2ekernel | autoload⟩
```

File c

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`

```
⟨*2ekernel⟩
\def\fmtname{LaTeX2e}
\edef\fmtversion{2011/06/27}
```

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:-)).

```
\iffalse
\def\reserved@a#1/#2/#3\@nil{%
  \count@\year
  \advance\count@-#1\relax
  \multiply\count@ by 12\relax
  \advance\count@\month
  \advance\count@-#2\relax}
\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.

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

This startup banner may be further modified by the code in `ltfinal.dtx` if a patch file is present.

```
\everyjob{\typeout{\fmtname
⟨autoload⟩\space(autoload version)%
                                       \space<\fmtversion>}}
\immediate\write16{\fmtname
⟨autoload⟩\space(autoload version)%
                                       \space<\fmtversion>}
⟨/2ekernel⟩
```

File d

ltdefns.dtx

11 Definitions

This section contains commands used in defining other macros.

```
⟨*2ekernel⟩
```

11.1 Initex initialisations

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

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

`\typeout` Display something on the terminal.

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

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

```
\newlinechar`\^^J
```

11.2 Saved versions of TEX primitives

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

`\@@par`

```
\let\@@par=\par
%\let\@@input=\input    %%% moved earlier
%\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 TEX 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@
            \defaulthyphenchar
          \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 LATEX will make `\-` effectively an encoding specific text command.

```
\let\@@hyph=\-        % Save original primitive definition
\def\-{\discretionary{-}{}{}}
```

\@dischyph

```
\let\@dischyph=\-
```

\@@italiccorr Save the original italic correction.

```
\let\@@italiccorr=\/
```

\@height
\@depth
\@width
\@minus
\@plus

The following definitions save token space. E.g., using `\@height` instead of height saves 5 tokens at the cost in time of one macro expansion.

```
\def\@height{height} \def\@depth{depth} \def\@width{width}
\def\@minus{minus}
\def\@plus{plus}
```

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

```
\def\hb@xt@{\hbox to}
```

```
\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 arguments `[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 ⟨*TEXT*⟩. 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 *{⟨*\FOO*⟩}[⟨*i*⟩]{⟨*TEXT*⟩}
Same as `\newcommand`, except it checks if `\FOO` already defined.

\newenvironment *{⟨*FOO*⟩}[⟨*i*⟩]{⟨*DEF1*⟩}{⟨*DEF2*⟩}
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` `{`⟨*message*⟩`}`

Produces a warning message on the terminal.

`\typein` `{`⟨*message*⟩`}`

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

`\typein` `[`⟨`\`*CS*⟩`]{`⟨*MSG*⟩`}`

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

`\typein`

```
\def\typein{%
  \let\@typein\relax
  \@testopt\@xtypein\@typein}
```

```
\def\@xtypein[#1]#2{%
  \typeout{#2}%
  \advance\endlinechar\@M
  \read\@inputcheck to#1%
  \advance\endlinechar-\@M
  \@typein}
```

`\@namedef`

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

`\@nameuse`

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

`\@cons`

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

`\@car`
`\@cdr`

```
\def\@car#1#2\@nil{#1}
\def\@cdr#1#2\@nil{#2}
```

`\@carcube` `\@carcube T1 ... Tn\@nil` = `T1 T2 T3` , $n > 3$

```
\def\@carcube#1#2#3#4\@nil{#1#2#3}
```

`\@onlypreamble`
`\@preamblecmds`

This macro adds its argument to the list of commands stored in `\@preamblecmds` to be disabled after `\begin{document}`. These commands are redefined to generate `\@notprerr` at this point.

```
\def\@preamblecmds{}
\def\@onlypreamble#1{%
  \expandafter\gdef\expandafter\@preamblecmds\expandafter{%
        \@preamblecmds\do#1}}
\@onlypreamble\@onlypreamble
\@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.

```
\def\@star@or@long#1{%
  \@ifstar
   {\let\l@ngrel@x\relax#1}%
   {\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.

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

`\newcommand` User level `\newcommand`.

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

`\new@command`

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

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

`\@argdef`
`\@xargdef`

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

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

Handle the second optional argument.

```
\long\def\@xargdef#1[#2][#3]#4{%
  \@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 exisiting command names. “Default” is the contents of the second optional argument of `(re)newcommand`.

The `\aut@global` command below is only used in the autoload format. If it is `\global` then a global definition will be made.

```
⟨autoload⟩\aut@global
     \expandafter\def\expandafter#1\expandafter{%
          \expandafter
          \@protected@testopt
          \expandafter
          #1%
          \csname\string#1\endcsname
          {#3}}%
```

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

```
       \expandafter\@yargdef
          \csname\string#1\endcsname
           \tw@
           {#2}%
           {#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.

```
\long\def\@testopt#1#2{%
  \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.

```
\def\@protected@testopt#1{%%
  \ifx\protect\@typeset@protect
    \expandafter\@testopt
  \else
    \@x@protect#1%
  \fi}
```

`\@yargdef`
`\@yargd@f`

These generate a primitive argument specification, from a LaTeX `[`⟨*digit*⟩`]` form; in fact ⟨*digit*⟩ can be anything such that `\number` ⟨*digit*⟩ is single digit.

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

Recall that LaTeX2.09 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`.

```
\long \def \@yargdef #1#2#3{%
  \ifx#2\tw@
    \def\reserved@b##11{[####1]}%
  \else
    \let\reserved@b\@gobble
  \fi
  \expandafter
    \@yargd@f \expandafter{\number #3}#1%
}
```

The `\aut@global` command below is only used in the autoload format. If it is `\global` then a global definition will be made.

```
\long \def \@yargd@f#1#2{%
  \def \reserved@a ##1#1##2##{%
⟨autoload⟩\aut@global
    \expandafter\def\expandafter#2\reserved@b ##1#1%
    }%
  \l@ngrel@x \reserved@a 0##1##2##3##4##5##6##7##8##9###1%
}
```

`\@reargdef`

```
\long\def\@reargdef#1[#2]{%
  \@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 `\@temp`*a*–*e*.)

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

`\renew@command`

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

`\@ifdefinable`
`\@@ifdefinable`
`\@rc@ifdefinable`

Test is user is allowed to define a command.

```
\long\def\@ifdefinable #1#2{%
     \edef\reserved@a{\expandafter\@gobble\string #1}%
     \@ifundefined\reserved@a
         {\edef\reserved@b{\expandafter\@carcube \reserved@a xxx\@nil}%
          \ifx \reserved@b\@qend \@notdefinable\else
            \ifx \reserved@a\@qrelax \@notdefinable\else
              #2%
            \fi
          \fi}%
         \@notdefinable}
```

Saved definition of `\@ifdefinable`.

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

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

```
\long\def\@rc@ifdefinable#1#2{%
  \let\@ifdefinable\@@ifdefinable
  #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.

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

`\new@environment`

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

`\@newenva`

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

`\@newenvb`

```
\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.

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

`\renew@environment`

```
\def\renew@environment#1{%
  \@ifundefined{#1}%
     {\@latex@error{Environment #1 undefined}\@ehc
     }\relax
  \expandafter\let\csname#1\endcsname\relax
```

```
⟨autoload⟩\aut@global
  \expandafter\let\csname end#1\endcsname\relax
  \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 `\par`s)

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

```
\long\def\@newenv#1#2#3#4{%
  \@ifundefined{#1}%
    {\expandafter\let\csname#1\expandafter\endcsname
                         \csname end#1\endcsname}%
    \relax
  \expandafter\new@command
     \csname #1\endcsname#2{#3}%
```

```
⟨autoload⟩\aut@global
     \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`.

```
\def\newif#1{%
  \count@\escapechar \escapechar\m@ne
⟨autoload⟩\aut@global
    \let#1\iffalse
    \@if#1\iftrue
    \@if#1\iffalse
  \escapechar\count@}
```

`\@if`

```
\def\@if#1#2{%
⟨autoload⟩\aut@global
  \expandafter\def\csname\expandafter\@gobbletwo\string#1%
                    \expandafter\@gobbletwo\string#2\endcsname
                       {\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.

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

`\provide@command`

```
\def\provide@command#1{%
  \begingroup
    \escapechar\m@ne\xdef\@gtempa{{\string#1}}%
  \endgroup
  \expandafter\@ifundefined\@gtempa
    {\def\reserved@a{\new@command#1}}%
    {\def\reserved@a{\renew@command\reserved@a}}%
   \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 befor 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.

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

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

```
\@onlypreamble\CheckCommand
```

`\check@command`

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

`\@check@c`

`\CheckCommand` itself just grabs all the arguments we need, without actually looking for `[` optional argument forms. Now define `\reserved@a`. If `\\reserved@a` is then defined, compare it with the “`\#1`’ otherwise compare `\reserved@a` with `#1`.

```
\long\def\@check@c#1#2#3{%
  \expandafter\let\csname\string\reserved@a\endcsname\relax
  \renew@command\reserved@a#2{#3}%
  \@ifundefined{\string\reserved@a}%
   {\@check@eq#1\reserved@a}%
   {\expandafter\@check@eq
            \csname\string#1\expandafter\endcsname
            \csname\string\reserved@a\endcsname}}
\@onlypreamble\@check@c
```

`\@check@eq`

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

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

`\@gobble`
`\@gobbletwo`
`\@gobblefour`

The `\@gobble` macro is used to get rid of its argument.

```
\long\def \@gobble #1{}
\long\def \@gobbletwo #1#2{}
\long\def \@gobblefour #1#2#3#4{}
```

`\@firstofone`
`\@firstoftwo`
`\@secondoftwo`

Some argument-grabbers.

```
\long\def\@firstofone#1{#1}
\long\def\@firstoftwo#1#2{#1}
\long\def\@secondoftwo#1#2{#2}
```

`\@iden`

`\@iden` is another name for `\@firstofone` for compatibility reasons.

```
\let\@iden\@firstofone
```

`\@thirdofthree`

Another grabber now used in the encoding specific section.

```
\long\def\@thirdofthree#1#2#3{#3}
```

`\@expandtwoargs`

A macro to totally expand two arguments to another macro

```
\def\@expandtwoargs#1#2#3{%
\edef\reserved@a{\noexpand#1{#2}{#3}}\reserved@a}
```

`\@backslashchar`

A category code 12 backslash.

```
\edef\@backslashchar{\expandafter\@gobble\string\\}
```

11.4 Robust commands and protect

Fragile and robust commands are one of the thornier issues in LaTeX’s commands. Whilst typesetting documents, LaTeX makes use of many of TeX’s features, such as arithmetic, defining macros, and setting variables. However, there are (at least) three different ocassions when these commands are not safe. These are called ‘moving arguments’ by LaTeX, 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 LaTeX 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 `\edef`s, `\mark`s 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`
`\@unexpandable@noexpand`

These commands are used for setting `\protect` inside `\edef`s.

```
\def\@unexpandable@protect{\noexpand\protect\noexpand}
%\def\@unexpandable@noexpand{\noexpand\noexpand\noexpand}
```

`\DeclareRobustCommand`
`\declare@robustcommand`

This is a package-writers command, which has the same syntax as `\newcommand`, 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 `\ifmmode`s inside `\halign`s. 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`.

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

```
\def\declare@robustcommand#1{%
   \ifx#1\@undefined\else\ifx#1\relax\else
      \@latex@info{Redefining \string#1}%
   \fi\fi
   \edef\reserved@a{\string#1}%
   \def\reserved@b{#1}%
   \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
⟨autoload⟩\aut@global
   \edef#1{%
      \ifx\reserved@a\reserved@b
         \noexpand\x@protect
         \noexpand#1%
      \fi
      \noexpand\protect
      \expandafter\noexpand\csname
         \expandafter\@gobble\string#1 \endcsname
   }%
   \let\@ifdefinable\@rc@ifdefinable
   \expandafter\new@command\csname
      \expandafter\@gobble\string#1 \endcsname
}
```

`\@x@protect`
`\x@protect`

```
\def\x@protect#1{%
   \ifx\protect\@typeset@protect\else
      \@x@protect#1%
   \fi
}
\def\@x@protect#1\fi#2#3{%
   \fi\protect#1%
}
```

`\@typeset@protect`

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

`\set@display@protect`
`\set@typeset@protect`

These macros set `\protect` appropriately for typesetting or displaying.

```
\def\set@display@protect{\let\protect\string}
\def\set@typeset@protect{\let\protect\@typeset@protect}
```

`\protected@edef`
`\protected@xdef`
`\unrestored@protected@xdef`
`\restore@protect`

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

```
\def\protected@edef{%
   \let\@@protect\protect
   \let\protect\@unexpandable@protect
   \afterassignment\restore@protect
   \edef
}
\def\protected@xdef{%
   \let\@@protect\protect
   \let\protect\@unexpandable@protect
   \afterassignment\restore@protect
   \xdef
}
\def\unrestored@protected@xdef{%
   \let\protect\@unexpandable@protect
   \xdef
}
\def\restore@protect{\let\protect\@@protect}
```

`\protect`

The normal meaning of `\protect`

```
\set@typeset@protect
```

11.5 Internal defining commands

These commands are used internally to define other LaTeX commands.

`\@ifundefined`

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

```
\def\@ifundefined#1{%
  \expandafter\ifx\csname#1\endcsname\relax
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
```

`\@qend`
`\@qrelax`

The following define `\@qend` and `\@qrelax` to be the strings '`end`' and '`relax`' with the characters `\catcode`d 12.

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

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

```
\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.

```
\def\@ifnch{%
  \ifx\@let@token\@sptoken
    \let\reserved@c\@xifnch
  \else
    \ifx\@let@token\reserved@d
      \let\reserved@c\reserved@a
    \else
      \let\reserved@c\reserved@b
    \fi
  \fi
  \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.

```
\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.

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

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

```
\def\makeatletter{\catcode`\@11\relax}
\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`.

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

`\@dblarg` `\@xdblarg`

```
\long\def\@dblarg#1{\kernel@ifnextchar[{#1}{\@xdblarg{#1}}}
\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.

```
\def\@sanitize{\@makeother\ \@makeother\\\@makeother\$\@makeother\&%
\@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.

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

```
⟨/2ekernel⟩
```

11.6 Commands for Autoloading

```
⟨*autoload⟩
```

`\aut@global` This command is only defined in the 'autoload' format. It is normally `\relax` but may be set to `\global`, in which case `\newif` and the commands based on `\newcommand` will all make global definitions.

```
\let\aut@global\relax
```

`\@autoload` This macro is only defined in the 'autoload' format. It inputs a package '`auto#1.sty`' within a local group, and with normalised catcodes. `\aut@global` is set to `\global` so that `\newif` `\newcommand` and related commands make global definitions.

```
\def\@autoload#1{%
  \begingroup
  \makeatletter
  \let\aut@global\global
  \nfss@catcodes
  \catcode`\ =10
  \let\@latex@e@error\@gobble
  \@@input auto#1.sty\relax
  \endgroup}
```

```
⟨/autoload⟩
```

File e

ltalloc.dtx

12 Counters

This section deals with counter and other variable allocation.

```
⟨*2ekernel⟩
```

The following are from plain TeX:

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

`\@ne` The number 1.

`\m@ne` The number −1.

`\tw@` The number 2.

`\sixt@@n` The number 16.

`\@m` The number 1000.

`\@MM` The number 20000.

`\@xxxii` The constant 32.

```
\chardef\@xxxii=32
```

`\@Mi` `\@Mii` `\@Miii` `\@miv` Constants 1001–1004.

```
\mathchardef\@Mi=10001
\mathchardef\@Mii=10002
\mathchardef\@Miii=10003
\mathchardef\@Miv=10004
```

`\@tempcnta` `\@tempcntb` Scratch count registers used by LaTeX kernel commands.

```
\newcount\@tempcnta
\newcount\@tempcntb
```

`\if@tempswa` General boolean switch used by LaTeX kernel commands.

```
\newif\if@tempswa
```

`\@tempdima` `\@tempdimb` `\@tempdimc` Scratch dimen registers used by LaTeX kernel commands.

```
\newdimen\@tempdima
\newdimen\@tempdimb
\newdimen\@tempdimc
```

`\@tempboxa` Scratch box register used by LaTeX kernel commands.

```
\newbox\@tempboxa
```

`\@tempskipa` `\@tempskipb` Scratch skip registers used by LaTeX kernel commands.

```
\newskip\@tempskipa
\newskip\@tempskipb
```

`\@temptokena` Scratch token register used by LaTeX kernel commands.

```
\newtoks\@temptokena
```

`\@flushglue` Glue used for `\right-` & `\leftskip` = 0pt plus 1fil

```
\newskip\@flushglue \@flushglue = 0pt plus 1fil
```

```
⟨/2ekernel⟩
```

File f

ltcntrl.dtx

13 Program control structure

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

```
⟨*2ekernel⟩
\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`
`\@iwhilenum`

```
\long\def\@whilenum#1\do #2{\ifnum #1\relax #2\relax\@iwhilenum{#1\relax
     #2\relax}\fi}
\long\def\@iwhilenum#1{\ifnum #1\expandafter\@iwhilenum
         \else\expandafter\@gobble\fi{#1}}
```

`\@whiledim`
`\@iwhiledim`

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

`\@whileswnoop`

`\@whilesw`
`\@iwhilesw`

```
\long\def\@whilesw#1\fi#2{#1#2\@iwhilesw{#1#2}\fi\fi}
\long\def\@iwhilesw#1\fi{#1\expandafter\@iwhilesw
         \else\@gobbletwo\fi{#1}\fi}
```

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

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

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

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

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

`\@nnil`

```
\def\@nnil{\@nil}
```

`\@empty`

```
\def\@empty{}
```

`\@fornoop`

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

`\@for`

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

`\@forloop`

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

`\@iforloop`

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

`\@tfor`

```
\def\@tfor#1:={\@tf@r#1 }
\long\def\@tf@r#1#2\do#3{\def\@fortmp{#2}\ifx\@fortmp\space\else
    \@tforloop#2\@nil\@nil\@@#1{#3}\fi}
\long\def\@tforloop#1#2\@@#3#4{\def#3{#1}\ifx #3\@nnil
      \expandafter\@fornoop \else
     #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.

```
\long\def\@break@tfor#1\@@#2#3{\fi\fi}
```

`\@removeelement` Removes an element from a comma-separated list and puts it into a control sequence, called as `\@removeelement{`⟨*element*⟩`}{`⟨*list*⟩`}{`⟨*cs*⟩`}`.

```
\def\@removeelement#1#2#3{%
  \def\reserved@a##1,#1,##2\reserved@a{##1,##2\reserved@b}%
  \def\reserved@b##1,\reserved@b##2\reserved@b{%
    \ifx,##1\@empty\else##1\fi}%
  \edef#3{%
    \expandafter\reserved@b\reserved@a,#2,\reserved@b,#1,\reserved@a}}
```

⟨/2ekernel⟩

File g
lterror.dtx

14 Error handling

This section defines LaTeX's error commands.

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

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

```
⟨*2ekernel | autoload⟩
```

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`.

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

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

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

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

```
\DeclareRobustCommand{\GenericWarning}[2]{%
   \begingroup
      \def\MessageBreak{^^J#1}%
      \set@display@protect
      \immediate\write\@unused{^^J#2\on@line.^^J}%
   \endgroup
}
⟨/2ekernel | autoload⟩
```

`\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.

```
⟨autoload⟩\def\GenericError{\@autoerr\GenericError}
⟨*2ekernel | def⟩
\bgroup
\lccode`\@=`\ %
\lccode`\~=`\ %
\lccode`\}=`\ %
\lccode`\{=`\ %
\lccode`\T=`\T%
\lccode`\H=`\H%
\catcode`\ =11\relax%
\lowercase{%
\egroup%
```

Unfortunately TeX versions older than 3.141 have a bug which means that `^^J` does not force a linebreak in `\message` and `\errmessage` commands. So for these old TeX'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 TeX! 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 TeX. See the comments in `ltdircheck.dtx`.

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

First the 'standard case'.

```
\DeclareRobustCommand{\GenericError}[4]{%
\begingroup%
\immediate\write\@unused{}%
\def\MessageBreak{^^J}%
\set@display@protect%
\edef%
%   %<--------------------do not delete this space!--------------------->%
\@err@                                                                   %
{{#4}}%
\errhelp
%   %<--------------------do not delete this space!--------------------->%
\@err@                                                                   %
\let
%    %<-------------------do not delete this space!--------------------->%
\@err@                                                                   %
\@empty
\def\MessageBreak{^^J#1}%
\def~{\errmessage{%
#2.^^J^^J%
#3^^J%
Type  H <return>  for immediate help%
%   %<--------------------do not delete this space!--------------------->%
\@err@                                                                   %
}}%
~%
\endgroup}%
```

```
\else%
```

Secondly the version for old TeX's.

```
\DeclareRobustCommand{\GenericError}[4]{%
\begingroup%
\immediate\write\@unused{}%
\def\MessageBreak{^^J}%
\set@display@protect%
\edef%
%   %<--------------------do not delete this space!--------------------->%
\@err@                                                                   %
{{#4}}%
\errhelp
%    %<-------------------do not delete this space!--------------------->%
\@err@                                                                   %
\let
%    %<-------------------do not delete this space!--------------------->%
\@err@                                                                   %
\errmessage
\def\MessageBreak{^^J#1}%
\def~{\typeout{! %
#2.^^J^^J%
#3^^J%
Type  H <return>  for immediate help.}%
%    %<-------------------do not delete this space!------------------->%
\@err@                                                                 %
{}}%
~%
\endgroup}%

\fi}%
⟨/2ekernel | def⟩
```

\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}{FOO}
\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 FOO on input line 54.
```

```
⟨autoload⟩\def\PackageError{\@autoerr\PackageError}
⟨*2ekernel | def⟩
\gdef\PackageError#1#2#3{%
   \GenericError{%
      (#1)\@spaces\@spaces\@spaces\@spaces
   }{%
      Package #1 Error: #2%
   }{%
      See the #1 package documentation for explanation.%
   }{#3}%
}
⟨/2ekernel | def⟩

⟨*2ekernel | autoload⟩
\def\PackageWarning#1#2{%
   \GenericWarning{%
      (#1)\@spaces\@spaces\@spaces\@spaces
   }{%
      Package #1 Warning: #2%
   }%
}
\def\PackageWarningNoLine#1#2{%
   \PackageWarning{#1}{#2\@gobble}%
}
\def\PackageInfo#1#2{%
```

```
    \GenericInfo{%
       (#1) \@spaces\@spaces\@spaces
    }{%
       Package #1 Info: #2%
    }%
}
⟨/2ekernel | autoload⟩

⟨autoload⟩\def\ClassError{\@autoerr\ClassError}
⟨*2ekernel | def⟩
\gdef\ClassError#1#2#3{%
    \GenericError{%
       (#1) \space\@spaces\@spaces\@spaces
    }{%
       Class #1 Error: #2%
    }{%
       See the #1 class documentation for explanation.%
    }{#3}%
}
⟨/2ekernel | def⟩

⟨*2ekernel | autoload⟩
\def\ClassWarning#1#2{%
    \GenericWarning{%
       (#1) \space\@spaces\@spaces\@spaces
    }{%
       Class #1 Warning: #2%
    }%
}
\def\ClassWarningNoLine#1#2{%
    \ClassWarning{#1}{#2\@gobble}%
}
\def\ClassInfo#1#2{%
    \GenericInfo{%
       (#1) \space\space\@spaces\@spaces
    }{%
       Class #1 Info: #2%
    }%
}
⟨/2ekernel | autoload⟩
```

`\@latex@error`
`\@latex@warning`
`\@latex@warning@no@line`
`\@latex@info`
`\@latex@info@no@line`

Errors and other info, for use in the LaTeX core.

```
⟨autoload⟩\def\@latex@error{\@autoerr\@latex@error}
⟨*2ekernel | def⟩
\gdef\@latex@error#1#2{%
    \GenericError{%
       \space\space\space\@spaces\@spaces\@spaces
    }{%
       LaTeX Error: #1%
    }{%
       See the LaTeX manual or LaTeX Companion for explanation.%
    }{#2}%
}
⟨/2ekernel | def⟩

⟨*2ekernel | autoload⟩
\def\@latex@warning#1{%
    \GenericWarning{%
       \space\space\space\@spaces\@spaces\@spaces
    }{%
       LaTeX Warning: #1%
    }%
}
```

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

```
\def\@latex@info#1{%
   \GenericInfo{%
      \@spaces\@spaces\@spaces
   }{%
      LaTeX Info: #1%
   }%
}
```

```
\def\@latex@info@no@line#1{%
  \@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 :-)

```
\let\c@errorcontextlines\errorcontextlines
\c@errorcontextlines=-1
```

`\on@line`

The message ‘ on input line *n*’, if possible.

```
\ifnum\inputlineno=\m@ne
  \let\on@line\@empty
\else
  \def\on@line{ on input line \the\inputlineno}
\fi
```

`\@warning`
`\@@warning`
`\@latexerr`

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

```
\let\@warning\@latex@warning
\let\@@warning\@latex@warning@no@line
⟨/2ekernel | autoload⟩
\global\let\@latexerr\@latex@error
```

`\@spaces`

Four spaces.

```
⟨*2ekernel | autoload⟩
\def\@spaces{\space\space\space\space}
⟨/2ekernel | autoload⟩
```

14.2 Specific errors

`\@eha`
`\@ehb`
`\@ehc`
`\@ehd`

The more common error help messages.

```
⟨*2ekernel | def⟩
\gdef\@eha{%
  Your command was ignored.\MessageBreak
  Type \space I <command> <return> \space to replace it %
  with another command,\MessageBreak
  or \space <return> \space to continue without it.}
\gdef\@ehb{%
  You’ve lost some text. \space \@ehc}
\gdef\@ehc{%
  Try typing \space <return> %
  \space to proceed.\MessageBreak
  If that doesn’t work, type \space X <return> \space to quit.}
\gdef\@ehd{%
  You’re in trouble here.  \space\@ehc}
⟨/2ekernel | def⟩
```

As `\latex@error` triggers the autoload, these definitions should not be needed in the autoload format, but just to be safe...

```
⟨*autoload⟩
\let\@eha\@empty\let\@ehb\@empty\let\@ehc\@empty\let\@ehd\@empty
⟨/autoload⟩
```

Here are most of the error message-generating commands of LaTeX.

`\@autoerr` Make this autoload command robust, as it may be read in at unpredicatble times.

```
⟨autoload⟩\def\@autoerr{\protect\@autoload{err}\protect}
```

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

```
\gdef\@notdefinable{%
⟨!autoload⟩ \@latex@error{%
⟨!autoload⟩   Command \@backslashchar\reserved@a\space
⟨!autoload⟩   already defined.\MessageBreak
⟨!autoload⟩   Or name \@backslashchar\@qend... illegal,
⟨!autoload⟩   see p.192 of the manual}\@eha}
⟨autoload⟩  \@autoerr\@notdefinable}
```

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

```
\gdef\@nolnerr{%
⟨!autoload⟩  \@latex@error{There’s no line here to end}\@eha}
⟨autoload⟩  \@autoerr\@nolnerr}
```

`\@nocounterr` Generated by `\setcounter`, `\addtocounter` or `\newcounter` if applied to an undefined counter ⟨*cnt*⟩.

`\@nocnterr` Obsolete error message generated in LaTeX2.09 by `\setcounter`, `\addtocounter` or `\newcounter` for undefined counter. DO NOT use for LaTeX 2$_\varepsilon$ it MIGHT vanish! Use `\@nocounterr{`⟨*cnt*⟩`}` instead.

```
\gdef\@nocounterr#1{%
⟨!autoload⟩  \@latex@error{No counter ’#1’ defined}\@eha}
⟨autoload⟩  \@autoerr\@nocounterr}
\gdef\@nocnterr{\@nocounterr?}
```

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

```
\gdef\@ctrerr{%
⟨!autoload⟩  \@latex@error{Counter too large}\@ehb}
⟨autoload⟩  \@autoerr\@ctrerr}
```

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

```
⟨!def⟩\gdef\@nodocument{%
⟨!def⟩  \@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.

```
\gdef\@badend#1{%
⟨!autoload⟩  \@latex@error{\protect\begin{\@currenvir}\@currenvline
⟨!autoload⟩                        \space ended by \protect\end{#1}}\@eha}
⟨autoload⟩  \@autoerr\@badend}
```

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

```
\gdef\@badmath{%
⟨!autoload⟩  \@latex@error{Bad math environment delimiter}\@eha}
⟨autoload⟩  \@autoerr\@badmath}
```

`\@toodeep` Called by a list environment nested more than six levels deep, or an enumerate or itemize nested more than four levels.

```
\gdef\@toodeep{%
⟨!autoload⟩  \@latex@error{Too deeply nested}\@ehd}
⟨autoload⟩  \@autoerr\@toodeep}
```

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

```
\gdef\@badpoptabs{%
⟨!autoload⟩  \@latex@error{\protect\pushtabs\space and \protect\poptabs
⟨!autoload⟩      \space don't match}\@ehd}
⟨autoload⟩  \@autoerr\@badpoptabs}
```

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

```
\gdef\@badtab{%
⟨!autoload⟩ \@latex@error{Undefined tab position}\@ehd}
⟨autoload⟩  \@autoerr\@badtab}
```

`\@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`.

```
\gdef\@preamerr#1{%
  \begingroup
    \let\protect\relax
⟨*!autoload⟩
    \@latex@error{\ifcase #1 Illegal character\or
     Missing @-exp\or Missing p-arg\fi\space
     in array arg}\@ehd
⟨/!autoload⟩
⟨autoload⟩  \@autoerr\@preamerr{#1}%
  \endgroup}
```

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

```
\gdef\@badlinearg{%
⟨!autoload⟩  \@latex@error{%
⟨!autoload⟩       Bad \protect\line\space or \protect\vector
⟨!autoload⟩       \space argument}\@ehb}
⟨autoload⟩  \@autoerr\@badlinearg}
```

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

```
\gdef\@parmoderr{%
⟨!autoload⟩  \@latex@error{Not in outer par mode}\@ehb}
⟨autoload⟩  \@autoerr\@parmoderr}
```

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

```
\gdef\@fltovf{%
⟨!autoload⟩  \@latex@error{Too many unprocessed floats}\@ehb}
⟨autoload⟩  \@autoerr\@fltovf}
```

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

```
\gdef\@latexbug{%
⟨!autoload⟩  \@latex@error{This may be a LaTeX bug}{Call for help}}
⟨autoload⟩  \@autoerr\@latexbug}
```

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

```
%\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`.

```
\gdef\@noitemerr{%
⟨!autoload⟩  \@latex@error{Something's wrong--perhaps a missing %
⟨!autoload⟩      \protect\item}\@ehc}
⟨autoload⟩  \@autoerr\@noitemerr}
```

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

```
\gdef\@notprerr{%
⟨!autoload⟩  \@latex@error{Can be used only in preamble}\@eha}
⟨autoload⟩  \@autoerr\@notprerr}
```

`\@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.

```
\gdef\@inmatherr#1{%
   \relax
   \ifmmode
⟨!autoload⟩ \@latex@error{Command \protect#1 invalid in math mode}\@ehc
⟨autoload⟩  \@autoerr\@inmatherr#1%
   \fi}
```

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

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

As well as the above error commands some error messages are directly coded to save space. The Messages alrerady present in LaTeX2.09 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.

```
⟨*2ekernel⟩
\message{par,}
```

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

`\@par`

```
\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.

```
\def\@par{\let\par\@@par\par}
⟨/2ekernel⟩
```

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

```
\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 `\vadjust{\penalty 10000}`
OBSOLETE COMMANDS (which never made it into the manual):
`\obeycr` : defines ¡CR¿ == `\\\relax`
`\restorecr` : restores ¡CR¿ 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 `\unskip`s'. 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 `\vadjust`s 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(?) `\unskip`s 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 `\unskip`s.
- Investigate and rationalise the 'newline' code.
- Find better algorithms for all sorts of things or, easier(?), fix TEX itself.

16.4 The code

```
⟨*2ekernel⟩
\message{spacing,}
```

`\pagebreak`
`\nopagebreak`

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

`\@no@pgbk`

```
\def\@no@pgbk #1[#2]{%
  \ifvmode
    \penalty #1\@getpen{#2}%
  \else
    \@bsphack
    \vadjust{\penalty #1\@getpen{#2}}%
    \@esphack
  \fi}
```

`\linebreak`
`\nolinebreak`

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

`\@no@lnbk`

```
\def\@no@lnbk #1[#2]{%
  \ifvmode
    \@nolnerr
  \else
    \@tempskipa\lastskip
    \unskip
    \penalty #1\@getpen{#2}%
    \ifdim\@tempskipa>\z@
      \hskip\@tempskipa
      \ignorespaces
    \fi
  \fi}
```

`\samepage`

```
\def\samepage{\interlinepenalty\@M
   \postdisplaypenalty\@M
   \interdisplaylinepenalty\@M
   \@beginparpenalty\@M
   \@endparpenalty\@M
   \@itempenalty\@M
   \@secpenalty\@M
   \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 `\\`.

```
\DeclareRobustCommand\\{%
  \let \reserved@e \relax
  \let \reserved@f \relax
  \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
            \@xnewline}%
          \@xnewline}
\expandafter\let\expandafter\@normalcr
     \csname\expandafter\@gobble\string\\ \endcsname
```

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

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

`\@xnewline`

```
\def\@xnewline{\@ifnextchar[% ] bracket matching
                  \@newline
                 {\@gnewline\relax}}
```

`\@newline`

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

```
\def\@gnewline #1{%
  \ifvmode
    \@nolnerr
  \else
    \unskip \reserved@e {\reserved@f#1}\nobreak \hfil \break
  \fi}
```

`\@getpen`

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

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

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

```
\def\@nobreakfalse{\global\let\if@nobreak\iffalse}
\def\@nobreaktrue {\global\let\if@nobreak\iftrue}
\@nobreakfalse
```

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

`\@savsf`

```
\newdimen\@savsk
\newcount\@savsf
```

`\@bsphack` `\@bsphack` and `\@esphack` used by macros such as `\index` and `\begin{@float}` . . . `\end{@float}` that want to be invisible — i.e., not leave any extra space when used in the middle of text. Such a macro should begin with `\@bsphack` and end with `\@esphack` The macro in question should not create any text, nor change the mode.

Before giving the current definition we give an extended definition that is currently not used (because it doesnt 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.

```
\def\@bsphack{%
  \relax
  \ifhmode
    \@savsk\lastskip
    \@savsf\spacefactor
  \fi}
```

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

```
\def\@esphack{%
  \relax
  \ifhmode
    \spacefactor\@savsf
    \ifdim\@savsk>\z@
      \ignorespaces
    \fi
  \fi}
```

`\@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.

```
\def\@Esphack{%
  \relax
  \ifhmode
    \spacefactor\@savsf
    \ifdim\@savsk>\z@
      \@ignoretrue
      \ignorespaces
    \fi
  \fi}
```

`\@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

LaTeX supports the plain TeX 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.

```
\def\@xaddvskip{%
```

```
  \ifdim\lastskip<\@tempskipb
    \vskip-\lastskip
    \vskip\@tempskipb
  \else
    \ifdim\@tempskipb<\z@
      \ifdim\lastskip<\z@
      \else
        \advance\@tempskipb\lastskip
        \vskip-\lastskip
        \vskip \@tempskipb
      \fi
    \fi
  \fi}
```

\addvspace

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

```
\def\addvspace#1{%
  \ifvmode
     \if@minipage\else
       \ifdim \lastskip =\z@
         \vskip #1\relax
       \else
       \@tempskipb#1\relax
         \@xaddvskip
       \fi
     \fi
  \else
    \@noitemerr
  \fi}
```

\addpenalty

```
\def\addpenalty#1{%
  \ifvmode
    \if@minipage
    \else
      \if@nobreak
      \else
        \ifdim\lastskip=\z@
          \penalty#1\relax
        \else
          \@tempskipb\lastskip
          \vskip -\lastskip
          \penalty#1%
          \vskip\@tempskipb
        \fi
      \fi
    \fi
  \else
    \@noitemerr
  \fi}
```

\vspace
\@vspace
\@vspacer

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

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

```
\DeclareRobustCommand\vspace{\@ifstar\@vspacer\@vspace}
\def\@vspace #1{%
  \ifvmode
```

```
    \vskip #1
    \vskip\z@skip
   \else
     \@bsphack
     \vadjust{\@restorepar
              \vskip #1
              \vskip\z@skip
              }%
     \@esphack
   \fi}
\def\@vspacer#1{%
  \ifvmode
    \dimen@\prevdepth
    \hrule \@height\z@
    \nobreak
    \vskip #1
    \vskip\z@skip
    \prevdepth\dimen@
  \else
    \@bsphack
    \vadjust{\@restorepar
             \hrule \@height\z@
             \nobreak
             \vskip #1
             \vskip\z@skip}%
    \@esphack
  \fi}
```

`\smallskip` `\medskip` `\bigskip`

```
\def\smallskip{\vspace\smallskipamount}
\def\medskip{\vspace\medskipamount}
\def\bigskip{\vspace\bigskipamount}
```

`\smallskipamount` `\medskipamount` `\bigskipamount`

```
\newskip\smallskipamount \smallskipamount=3pt plus 1pt minus 1pt
\newskip\medskipamount   \medskipamount  =6pt plus 2pt minus 2pt
\newskip\bigskipamount   \bigskipamount =12pt plus 4pt minus 4pt
```

16.6 Horizontal space (and breaks)

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

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

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

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!

```
\DeclareRobustCommand{\nobreakdashes}{%
  \leavevmode
  \toks@{}%
  \def\reserved@a##1{\toks@\expandafter{\the\toks@-}%
                     \futurelet\@let@token \reserved@b}%
  \def\reserved@b  {\ifx\@let@token -%
                       \expandafter\reserved@a
                     \else
                       \setbox\z@ \hbox{\the\toks@\nobreak}%
```

```
                        \unhbox\z@
                        \spacefactor\sfcode`\-
                      \fi}%
  \futurelet\@let@token \reserved@b
}
```

`\nobreakspace` `\@xobeysp` 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` .

```
\DeclareRobustCommand{\nobreakspace}{%
   \leavevmode\nobreak\ }
\catcode `\~=13
\def~{\nobreakspace{}}
\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.'' ```

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

`\@` 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.

```
\def\@{\spacefactor\@m}
```

`\hspace`

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

`\@hspace`

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

`\@hspacer` extra `\hskip 0pt` added 1985/17/12 to guard against a following `\unskip` `\relax` added 13 Oct 88 for usual TeX lossage replaced both changes by `\hskip\z@skip` 27 Nov 91

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

`\fill`

```
\newskip\fill
\fill = 0pt plus 1fill
```

`\stretch`

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

`\thinspace`

`\negthinspace` `\enspace`

```
\def\thinspace{\kern .16667em }
\def\negthinspace{\kern-.16667em }
\def\enspace{\kern.5em }
```

`\enskip`

`\quad` `\qquad`

```
\def\enskip{\hskip.5em\relax}
\def\quad{\hskip1em\relax}
\def\qquad{\hskip2em\relax}
```

`\obeycr`
`\restorecr`

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

```
{\catcode`\^^M=13 \gdef\obeycr{\catcode`\^^M13 \def^^M{\\\relax}%
    \@gobblecr}%
{\catcode`\^^M=13 \gdef\@gobblecr{\@ifnextchar
\@gobble\ignorespaces}}
\gdef\restorecr{\catcode`\^^M5 }}
```

⟨/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.

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

`\LaTeX` The LaTeX logo.

```
\DeclareRobustCommand{\LaTeX}{L\kern-.36em%
        {\sbox\z@ T%
         \vbox to\ht\z@{\hbox{\check@mathfonts
                              \fontsize\sf@size\z@
                              \math@fontsfalse\selectfont
                              A}%
                        \vss}%
        }%
        \kern-.15em%
        \TeX}
```

`\LaTeXe` The LaTeX 2$_\varepsilon$ logo as proposed by A-W designers.

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

File k

ltfiles.dtx

18 File Handling

The following user commands are defined in this part:

`\document` (ie `\begin{document}`)
Reads in the .AUX files and `\catcode`'s `@` to 12.

`\nofiles`
Suppresses all file output by setting `\@filesw` false.

`\includeonly` {⟨*NAME1, ... ,NAMEn*⟩}
Causes only parts NAME1, ... ,NAMEn to be read by their `\include` commands. Works by setting partsw true and setting `\@partlist` to NAME1, ... ,NAMEn.

`\include` {⟨*NAME*⟩}
Does an `\input` NAME unless `\@partsw` is true and NAME is not in `\@partlist`. If `\@filesw` is true, then it directs .AUX output to NAME.AUX, including a checkpoint at the end.

`\input` {⟨*NAME*⟩}
The same as TeX's `\input`, 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.

`\IfFileExists` {⟨*NAME*⟩}{⟨*then*⟩}{⟨*else*⟩}
If the file exists on the system, execute *then* otherwise execute *else*.

`\InputIfFileExists` {⟨*NAME*⟩}{⟨*then*⟩}{⟨*else*⟩}
If the file exists on the system, execute *then* and input *NAME* otherwise execute *else*.

```
⟨*2ekernel | autoload⟩
\message{files,}
```

```
VARIABLES, SWITCHES AND INTERNAL COMMANDS:
  \@mainaux    : Output file number for main .AUX file.
  \@partaux    : Output file number for current part's .AUX file.
  \@auxout     : Either \@mainout or \@partout, depending on
                   which .AUX file output goes to.
  \@input{foo} : If file foo exists, then \input's it,
                   otherwise types a warning message.
  @filesw      : Switch - set false if no .AUX, .TOC, .IDX etc
                     files are to be written
  @partsw      : Set true by a \includeonly command.
  \@partlist   : Set to the argument of the \includeonly command.

  \cp@FOO      : The checkpoint for \include'd file FOO.TEX, written
                   by \@writeckpt 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` `\@unused` Allocate read stream for testing and output stream.

```
\newread\@inputcheck
\newwrite\@unused
```

`\@mainaux` `\@partaux`

```
\newwrite\@mainaux
\newwrite\@partaux
```

`\if@filesw`

`\if@partsw`

```
\newif\if@filesw \@fileswtrue
\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).

```
\newcount\@clubpenalty
\@clubpenalty \clubpenalty
```

`\document` Cancel the `\begingroup` from `\begin`

```
\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.

```
\ifx\@unusedoptionlist\@empty\else
  \@latex@warning@no@line{Unused global option(s):^^J%
          \@spaces[\@unusedoptionlist]}%
\fi
\@colht\textheight
\@colroom\textheight \vsize\textheight
\columnwidth\textwidth
\@clubpenalty\clubpenalty
\if@twocolumn
  \advance\columnwidth -\columnsep
  \divide\columnwidth\tw@ \hsize\columnwidth \@firstcolumntrue
\fi
\hsize\columnwidth \linewidth\hsize
\begingroup\@floatplacement\@dblfloatplacement
  \makeatletter\let\@writefile\@gobbletwo

  \global \let \@multiplelabels \relax
  \@input{\jobname.aux}%
\endgroup
\if@filesw
  \immediate\openout\@mainaux\jobname.aux
  \immediate\write\@mainaux{\relax}%
\fi
```

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

```
\process@table
\let\glb@currsize\@empty  %% Force math initialization.

\normalsize
\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 LaTeX2.0x and plain TeX.)

```
\ifx\normalsfcodes\@empty
  \ifnum\sfcode`\.=\@m
    \let\normalsfcodes\frenchspacing
  \else
    \let\normalsfcodes\nonfrenchspacing
  \fi
\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.

```
  \@noskipsecfalse
```

```
  \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...

```
  \let\AtBeginDocument\@firstofone
  \@begindocumenthook
```

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

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

```
  \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
  \global\@maxdepth\maxdepth
  \global\let\@begindocumenthook\@undefined
  \ifx\@listfiles\@undefined
    \global\let\@filelist\relax
    \global\let\@addtofilelist\@gobble
  \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.

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

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

```
  \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.

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

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

```
  \ignorespaces}
```

```
\@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.

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

`\nofiles` Set `\@fileswfalse` which suppresses the places where LaTeX 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.

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

```
\long\def \protected@write#1#2#3{%
      \begingroup
       \let\thepage\relax
       #2%
       \let\protect\@unexpandable@protect
       \edef\reserved@a{\write#1{#3}}%
       \reserved@a
      \endgroup
      \if@nobreak\ifvmode\nobreak\fi\fi
}
```

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

`\includeonly`

```
\def\includeonly#1{%
  \@partswtrue
  \edef\@partlist{\zap@space#1 \@empty}}
\@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.

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

`\@include`

```
\def\@include#1 {%
  \clearpage
  \if@filesw
    \immediate\write\@mainaux{\string\@input{#1.aux}}%
  \fi
  \@tempswatrue
  \if@partsw
    \@tempswafalse
    \edef\reserved@b{#1}%
    \@for\reserved@a:=\@partlist\do
      {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
  \fi
  \if@tempswa
    \let\@auxout\@partaux
    \if@filesw
      \immediate\openout\@partaux #1.aux
      \immediate\write\@partaux{\relax}%
    \fi
    \@input@{#1.tex}%
    \clearpage
    \@writeckpt{#1}%
    \if@filesw
      \immediate\closeout\@partaux
    \fi
  \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.

```
    \deadcycles\z@
    \@nameuse{cp@#1}%
  \fi
  \let\@auxout\@mainaux}
```

`\@writeckpt`

```
\def\@writeckpt#1{%
  \if@filesw
    \immediate\write\@partaux{\string\@setckpt{#1}\@charlb}%
    {\let\@elt\@wckptelt \cl@@ckpt}%
    \immediate\write\@partaux{\@charrb}%
  \fi}
```

`\@wckptelt`

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

`\@setckpt` RmS 93/08/31: introduced `\@setckpt`

```
\def\@setckpt#1{\global\@namedef{cp@#1}}
```

`\@charlb` `\@charrb` The following defines `\@charlb` and `\@charrb` to be `{` and `}`, respectively with `\catcode` 11.

```
{\catcode`[=1 \catcode`]=2
\catcode`{=11 \catcode`}=11
\gdef\@charlb[{]
\gdef\@charrb[}]
]% }brace matching
```

18.1 Safe Input Macros

`\IfFileExists`

```
\long\def \IfFileExists#1#2#3{%
  \openin\@inputcheck#1 %
  \ifeof\@inputcheck
    \ifx\input@path\@undefined
      \def\reserved@a{#3}%
    \else
      \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
    \fi
  \else
    \closein\@inputcheck
    \edef\@filef@und{#1 }%
    \def\reserved@a{#2}%
  \fi
  \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`.

```
\long\def\@iffileonpath#1{%
  \let\reserved@a\@secondoftwo
  \expandafter\@tfor\expandafter\reserved@b\expandafter
```

```
              :\expandafter=\input@path\do{%
   \openin\@inputcheck\reserved@b#1 %
   \ifeof\@inputcheck\else
     \edef\@filef@und{\reserved@b#1 }%
     \let\reserved@a\@firstoftwo%
     \closein\@inputcheck
     \@break@tfor
   \fi}%
  \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`'.

```
\long\def \InputIfFileExists#1#2{%
  \IfFileExists{#1}%
    {#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).

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

`\@iinput`

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

```
\def\@iinput#1{%
  \InputIfFileExists{#1}{}%
  {\filename@parse{#1}%
   \edef\reserved@a{\noexpand\@missingfileerror
     {\filename@area\filename@base}%
     {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
   \reserved@a}}
```

`\@input`

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

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

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

`\@input@`

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

```
\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.

```
⟨autoload⟩\def\@missingfileerror{\@autoerr\@missingfileerror}
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoerr⟩
\gdef\@missingfileerror#1#2{%
     \typeout{^^J! LaTeX Error: File `#1.#2' not found.^^J^^J%
      Type X to quit or <RETURN> to proceed,^^J%
      or enter new name. (Default extension: #2)^^J}%
     \message{Enter file name: }%
      {\endlinechar\m@ne
       \global\read\m@ne to\@gtempa}%
    \ifx\@gtempa\@empty
    \else
      \def\reserved@a{x}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
      \def\reserved@a{X}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
      \filename@parse\@gtempa
      \edef\filename@ext{%
        \ifx\filename@ext\relax#2\else\filename@ext\fi}%
     \edef\reserved@a{%
       \noexpand\InputIfFileExists
```

```
         {\filename@area\filename@base.\filename@ext}%
         {}%
         {\noexpand\@missingfileerror
            {\filename@area\filename@base}{\filename@ext}}}%
      \reserved@a
    \fi}
⟨/2ekernel | autoerr⟩
⟨*2ekernel | autoload⟩
```

`\@obsoletefile` For compatibility with LaTeX 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.

```
\def\@obsoletefile#1#2{%
  \@latex@warning@no@line{inputting `#1' instead of obsolete `#2'}}
\@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.

```
\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.

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

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

```
\def\listfiles{%
  \let\listfiles\relax
  \def\@listfiles##1##2##3##4##5##6##7##8##9\@@{%
     \def\reserved@d{\\}%
     \@tfor\reserved@c:=##1##2##3##4##5##6##7##8\do{%
       \ifx\reserved@c\reserved@d
         \edef\filename@area{ \filename@area}%
       \fi}}%
```

```
  \def\@dofilelist{%
     \typeout{^^J *File List*}%
     \@for\@currname:=\@filelist\do{%
       \filename@parse\@currname
       \edef\reserved@a{%
          \filename@base.%
          \ifx\filename@ext\relax tex\else\filename@ext\fi}%
       \expandafter\let\expandafter\reserved@b
                              \csname ver@\reserved@a\endcsname
       \expandafter\expandafter\expandafter\@listfiles\expandafter
           \filename@area\filename@base\\\\\\\\\\\\\\\\\\\@@
       \typeout{%
          \filename@area\reserved@a
          \ifx\reserved@b\relax\else\@spaces\reserved@b\fi}}%
     \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}
```

```
\@onlypreamble\listfiles
```

`\@dofilelist`

```
\let\@dofilelist\relax
⟨/2ekernel | autoload⟩
```

File l

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 defintion 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 {\frac14}}
```

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` LaTeX will still find the encoding specific-definition for `OT1` and therefore ignore the new default. Therefore to ensure that in this case the `TS1` version is used we have to remove the `OT1` declaration:

```
\UndeclareTextCommand{\textdollar}{OT1}
```

Since the \$ sign is a proper glyph in the `T1` encoding there is no point removing its definition and forcing LaTeX 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 `t1enc.def` for the Cork encoding.
`TS1`	generates `ts1enc.def` for the Text Companion encoding.
`TS1sty`	generates `textcomp.sty`, package that sets up use of the Text Companion encoding.
`OT1`	generates `ot1enc.def` for Knuth's CM encoding.
`OMS`	generates `omsenc.def` for Knuth's math symbol encoding.
`OML`	generates `omlenc.def` for Knuth's math letters encoding.
`OT4`	generates `ot4enc.def` 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 `fontenc.sty` for selecting encodings.
`2ekernel`	for the kernel commands.
`autoload`	for the 'autoload' kernel commands.
`autoerr`	for the autoerr.sty error message autoload file.

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.

```
⟨*2ekernel | autoload⟩
\message{font encodings,}
```

Far too many macros in one block here!

`\DeclareTextCommand`
`\ProvideTextCommand`
`\DeclareTextSymbol`
`\@dec@text@cmd`
`\chardef@text@cmd`
`\@changed@cmd`
`\@changed@x`
`\TextSymbolUnavailable`
`\@inmathwarn`

If you say:

```
\DeclareTextCommand{\foo}{T1}...
```

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

```
\def\DeclareTextCommand{%
   \@dec@text@cmd\newcommand}

\def\ProvideTextCommand{%
   \@dec@text@cmd\providecommand}

\def\@dec@text@cmd#1#2#3{%
   \expandafter\def\expandafter#2%
      \expandafter{%
         \csname#3-cmd\expandafter\endcsname
         \expandafter#2%
         \csname#3\string#2\endcsname
      }%
   \let\@ifdefinable\@rc@ifdefinable
   \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.

```
\def\chardef@text@cmd{%
```

```
  \let\@ifdefinable\@@ifdefinable
  \chardef
 }
\def\DeclareTextSymbol#1#2#3{%
  \@dec@text@cmd\chardef@text@cmd#1{#2}#3\relax
 }
```

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

```
\@onlypreamble\DeclareTextCommand
\@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.

```
\def\@current@cmd#1{%
  \ifx\protect\@typeset@protect
    \@inmathwarn#1%
  \else
    \noexpand#1\expandafter\@gobble
  \fi}
```

```
\def\@changed@cmd#1#2{%
  \ifx\protect\@typeset@protect
    \@inmathwarn#1%
    \expandafter\ifx\csname\cf@encoding\string#1\endcsname\relax
      \expandafter\ifx\csname ?\string#1\endcsname\relax
        \expandafter\def\csname ?\string#1\endcsname{%
          \TextSymbolUnavailable#1%
        }%
      \fi
      \global\expandafter\let
          \csname\cf@encoding \string#1\expandafter\endcsname
          \csname ?\string#1\endcsname
    \fi
    \csname\cf@encoding\string#1%
```

```
         \expandafter\endcsname
   \else
      \noexpand#1%
   \fi}
```

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoerr⟩
\gdef\TextSymbolUnavailable#1{%
   \@latex@error{%
      Command \protect#1 unavailable in encoding \cf@encoding%
   }\@eha}
⟨/2ekernel | autoerr⟩
⟨autoload⟩\gdef\TextSymbolUnavailable{\@autoerr\TextSymbolUnavailable}
⟨*2ekernel | autoload⟩
```

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).

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

`\DeclareTextCommandDefault`
`\ProvideTextCommandDefault`

These define commands with encoding `?`.

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.

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

```
\def\ProvideTextCommandDefault#1{%
   \ProvideTextCommand#1?}
```

```
\@onlypreamble\DeclareTextCommandDefault
%\@onlypreamble\ProvideTextCommandDefault
```

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

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

`\DeclareTextAccent`

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

```
\def\DeclareTextAccent#1#2#3{%
   \DeclareTextCommand#1{#2}{\add@accent{#3}}}
```

```
\@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.

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

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

```
   \let\hmode@start@before@group\@firstofone
   \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.

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

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

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

`\hmode@bgroup`

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

`\DeclareTextCompositeCommand`
`\DeclareTextComposite`
`\@text@composite`
`\@text@composite@x`
`\@strip@args`

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:

```
#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`.

```
\def\DeclareTextCompositeCommand#1#2#3#4{%
   \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
   \expandafter\expandafter\expandafter\ifx
   \expandafter\@car\reserved@a\relax\relax\@nil \@text@composite \else
      \edef\reserved@b##1{%
         \def\expandafter\noexpand
            \csname#2\string#1\endcsname####1{%
            \noexpand\@text@composite
               \expandafter\noexpand\csname#2\string#1\endcsname
               ####1\noexpand\@empty\noexpand\@text@composite
               {##1}}}%
      \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
   \fi
   \expandafter\def\csname\expandafter\string\csname
      #2\endcsname\string#1-\string#3\endcsname{#4}}
```

```
\@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.

```
\def\@text@composite#1#2#3\@text@composite{%
   \expandafter\@text@composite@x
      \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 form the input stream. This has now being 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.

```
\def\@text@composite@x#1{%
   \ifx#1\relax
      \expandafter\@secondoftwo
   \else
      \expandafter\@firstoftwo
   \fi
   #1}
```

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

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

```
\def\DeclareTextComposite#1#2#3#4{%
   \def\reserved@a{\DeclareTextCompositeCommand#1{#2}{#3}}%
   \bgroup
      \lccode\z@#4%
      \lowercase{%
   \egroup
      \reserved@a ^^@}}
```

```
\catcode\z@=15\relax
```

```
\@onlypreamble\DeclareTextComposite
```

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

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

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

```
\def\UseTextAccent#1#2#3{%
  \hmode@start@before@group
  {%
```

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

```
     \let\hmode@start@before@group\@firstofone
     \let\@curr@enc\cf@encoding
     \@use@text@encoding{#1}%
```

```
    #2{\@use@text@encoding\@curr@enc#3}%
   }}
```

```
\def\UseTextSymbol#1#2{%
       \hmode@start@before@group
       {%
          \def\@wrong@font@char{\MessageBreak
             for \noexpand\symbol`\string#2'}%
          \@use@text@encoding{#1}%
          #2%
       }%
    }
```

```
\def\@use@text@encoding#1{%
   \edef\f@encoding{#1}%
   \xdef\font@name{%
      \csname\curr@fontshape/\f@size\endcsname}%
   \pickup@font
   \font@name
   \@@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.

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

`\DeclareTextSymbolDefault`
`\DeclareTextAccentDefault`

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

```
\def\DeclareTextSymbolDefault#1#2{%
   \DeclareTextCommandDefault#1{\UseTextSymbol{#2}#1}}
```

```
\def\DeclareTextAccentDefault#1#2{%
   \DeclareTextCommandDefault#1{\UseTextAccent{#2}#1}}
```

```
\@onlypreamble\DeclareTextSymbolDefault
\@onlypreamble\DeclareTextAccentDefault
```

`\UndeclareTextCommand`

This command safely removes and 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.

```
\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).

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

Else: throw away that declaration.

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

```
     \expandafter\expandafter\expandafter
     \ifx\expandafter\@thirdofthree#1\@undefined
       \expandafter\gdef\expandafter#1\expandafter
          {\csname ?-cmd\expandafter\endcsname\expandafter
           #1\csname?\string#1\endcsname}%
     \fi
 \fi
}
```

```
\@onlypreamble\UndeclareTextCommand
```

19.4.2 Hyphenation

`\patterns` `\@@patterns` `\hyphenation` `\@@hyphenation`

We redefine `\patterns` and `\hyphenation` to allow the use of commands declared with `\DeclareText*` to be used inside them.

```
%\let\@@patterns\patterns
%\let\@@hyphenation\hyphenation
%\def\patterns{%
%   \bgroup
%      \let\protect\@empty
%      \let\@typeset@protect\@empty
%      \let\@changed@x\@changed@x@mouth
%   \afterassignment\egroup
%   \@@patterns
%}
%\def\hyphenation{%
%   \bgroup
%      \let\protect\@empty
%      \let\@typeset@protect\@empty
%      \let\@changed@x\@changed@x@mouth
%   \afterassignment\egroup
%   \@@hyphenation
%}
```

19.4.3 Miscellania

`\a`

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

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

```
\def\@tabacckludge#1{\expandafter\@changed@cmd
                                 \csname\string#1\endcsname\relax}
\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 TEX 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:

```
\DeclareTextAccentDefault{\"}{OT1}
\DeclareTextAccentDefault{\'}{OT1}
\DeclareTextAccentDefault{\.}{OT1}
\DeclareTextAccentDefault{\=}{OT1}
\DeclareTextAccentDefault{\H}{OT1}
\DeclareTextAccentDefault{\^}{OT1}
\DeclareTextAccentDefault{\`}{OT1}
\DeclareTextAccentDefault{\b}{OT1}
\DeclareTextAccentDefault{\c}{OT1}
\DeclareTextAccentDefault{\d}{OT1}
\DeclareTextAccentDefault{\r}{OT1}
\DeclareTextAccentDefault{\u}{OT1}
\DeclareTextAccentDefault{\v}{OT1}
\DeclareTextAccentDefault{\~}{OT1}
```

Some symbols from OT1:

```
%\DeclareTextSymbolDefault{\AA}{OT1}
\DeclareTextSymbolDefault{\AE}{OT1}
\DeclareTextSymbolDefault{\L}{OT1}
\DeclareTextSymbolDefault{\OE}{OT1}
\DeclareTextSymbolDefault{\O}{OT1}
%\DeclareTextSymbolDefault{\aa}{OT1}
\DeclareTextSymbolDefault{\ae}{OT1}
\DeclareTextSymbolDefault{\i}{OT1}
\DeclareTextSymbolDefault{\j}{OT1}

\DeclareTextSymbolDefault{\ij}{OT1}
\DeclareTextSymbolDefault{\IJ}{OT1}

\DeclareTextSymbolDefault{\l}{OT1}
\DeclareTextSymbolDefault{\oe}{OT1}
\DeclareTextSymbolDefault{\o}{OT1}
\DeclareTextSymbolDefault{\ss}{OT1}
\DeclareTextSymbolDefault{\textdollar}{OT1}
\DeclareTextSymbolDefault{\textemdash}{OT1}
\DeclareTextSymbolDefault{\textendash}{OT1}
\DeclareTextSymbolDefault{\textexclamdown}{OT1}
%\DeclareTextSymbolDefault{\texthyphenchar}{OT1}
%\DeclareTextSymbolDefault{\texthyphen}{OT1}
\DeclareTextSymbolDefault{\textquestiondown}{OT1}
\DeclareTextSymbolDefault{\textquotedblleft}{OT1}
\DeclareTextSymbolDefault{\textquotedblright}{OT1}
\DeclareTextSymbolDefault{\textquoteleft}{OT1}
\DeclareTextSymbolDefault{\textquoteright}{OT1}
\DeclareTextSymbolDefault{\textsterling}{OT1}
```

Some symbols from OMS:

```
\DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
\DeclareTextSymbolDefault{\textbackslash}{OMS}
\DeclareTextSymbolDefault{\textbar}{OMS}
\DeclareTextSymbolDefault{\textbardbl}{OMS}
\DeclareTextSymbolDefault{\textbraceleft}{OMS}
\DeclareTextSymbolDefault{\textbraceright}{OMS}
\DeclareTextSymbolDefault{\textbullet}{OMS}
\DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
\DeclareTextSymbolDefault{\textdagger}{OMS}
```

```
\DeclareTextSymbolDefault{\textparagraph}{OMS}
\DeclareTextSymbolDefault{\textperiodcentered}{OMS}
\DeclareTextSymbolDefault{\textsection}{OMS}
\DeclareTextAccentDefault{\textcircled}{OMS}
```

Some symbols from OML:

```
\DeclareTextSymbolDefault{\textless}{OML}
\DeclareTextSymbolDefault{\textgreater}{OML}
\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.

```
\DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
% \expandafter\def\expandafter
%                 \copyright\expandafter{\expandafter{\copyright}}

\DeclareTextCommandDefault{\textasciicircum}{\^{}}
\DeclareTextCommandDefault{\textasciitilde}{\~{}}
\DeclareTextCommandDefault{\textcompwordmark}{\leavevmode\kern\z@}
\DeclareTextCommandDefault{\textunderscore}{%
  \leavevmode \kern.06em\vbox{\hrule\@width.3em}}

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

```
\DeclareTextCommandDefault{\textellipsis}{%
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font}

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

\DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
\DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}
```

19.4.5 Math material

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

```
\DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
\DeclareRobustCommand{\{}{\ifmmode\lbrace\else\textbraceleft\fi}
\DeclareRobustCommand{\}}{\ifmmode\rbrace\else\textbraceright\fi}
\DeclareRobustCommand{\P}{\ifmmode\mathparagraph\else\textparagraph\fi}
\DeclareRobustCommand{\S}{\ifmmode\mathsection\else\textsection\fi}
\DeclareRobustCommand{\dag}{\ifmmode{\dagger}\else\textdagger\fi}
\DeclareRobustCommand{\ddag}{\ifmmode{\ddagger}\else\textdaggerdbl\fi}
```

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

```
\DeclareRobustCommand{\_}{%
   \ifmmode\nfss@text{\textunderscore}\else\textunderscore\fi}
\DeclareRobustCommand{\copyright}{%
   \ifmmode{\nfss@text{\textcopyright}}\else\textcopyright\fi}
\DeclareRobustCommand{\pounds}{%
   \ifmmode\mathsterling\else\textsterling\fi}

\DeclareRobustCommand{\dots}{%
   \ifmmode\mathellipsis\else\textellipsis\fi}
```

```
\let\ldots\dots
⟨/2ekernel | autoload⟩
```

19.5 Definitions for the OT1 encoding

The definitions for the 'TEX text' (OT1) encoding.

Declare the encoding.

```
⟨*OT1⟩
\DeclareFontEncoding{OT1}{}{}
```

Declare the accents.

```
\DeclareTextAccent{\"}{OT1}{127}
\DeclareTextAccent{\'}{OT1}{19}
\DeclareTextAccent{\.}{OT1}{95}
\DeclareTextAccent{\=}{OT1}{22}
\DeclareTextAccent{\^}{OT1}{94}
\DeclareTextAccent{\`}{OT1}{18}
\DeclareTextAccent{\~}{OT1}{126}
\DeclareTextAccent{\H}{OT1}{125}
\DeclareTextAccent{\u}{OT1}{21}
\DeclareTextAccent{\v}{OT1}{20}
\DeclareTextAccent{\r}{OT1}{23}
```

Some accents have to be built by hand: Note that `\ooalign` and `\o@lign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from plain.tex since that now has two incompatible definitions.

```
\DeclareTextCommand{\b}{OT1}[1]
   {\hmode@bgroup\o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-3ex}%
     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
\DeclareTextCommand{\c}{OT1}[1]
   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
    \else{\ooalign{\unhbox\z@\crcr\hidewidth\char24\hidewidth}}\fi}
\DeclareTextCommand{\d}{OT1}[1]
   {\hmode@bgroup
    \o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
```

Declare the text symbols.

```
\DeclareTextSymbol{\AE}{OT1}{29}
\DeclareTextSymbol{\OE}{OT1}{30}
\DeclareTextSymbol{\O}{OT1}{31}
\DeclareTextSymbol{\ae}{OT1}{26}
\DeclareTextSymbol{\i}{OT1}{16}
\DeclareTextSymbol{\j}{OT1}{17}
\DeclareTextSymbol{\oe}{OT1}{27}
\DeclareTextSymbol{\o}{OT1}{28}
\DeclareTextSymbol{\ss}{OT1}{25}
\DeclareTextSymbol{\textemdash}{OT1}{124}
\DeclareTextSymbol{\textendash}{OT1}{123}
```

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

```
%\DeclareTextSymbol{\textexclamdown}{OT1}{60}
%\DeclareTextSymbol{\textquestiondown}{OT1}{62}
\DeclareTextCommand{\textexclamdown}{OT1}{!`}
\DeclareTextCommand{\textquestiondown}{OT1}{?`}
%\DeclareTextSymbol{\texthyphenchar}{OT1}{`\-}
%\DeclareTextSymbol{\texthyphen}{OT1}{`\-}
\DeclareTextSymbol{\textquotedblleft}{OT1}{92}
\DeclareTextSymbol{\textquotedblright}{OT1}{`\"}
\DeclareTextSymbol{\textquoteleft}{OT1}{`\`}
\DeclareTextSymbol{\textquoteright}{OT1}{`\'}
```

Some symbols which are faked from others:

```
% \DeclareTextCommand{\aa}{OT1}
%    {{\accent23a}}
\DeclareTextCommand{\L}{OT1}
   {\leavevmode\setbox\z@\hbox{L}\hb@xt@\wd\z@{\hss\@xxxii L}}
\DeclareTextCommand{\l}{OT1}
   {\hmode@bgroup\@xxxii l\egroup}
% \DeclareTextCommand{\AA}{OT1}
%    {\leavevmode\setbox\z@\hbox{h}\dimen@\ht\z@\advance\dimen@-1ex%
%     \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`.

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

The dutch language uses the letter ‘ij’. It is available in `T1` encoded fonts, but not in the `OT1` encoded fonts. Therefor we fake it for the `OT1` encoding.

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

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

```
\DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \slshape
   \else
      \upshape
   \fi
   \char`\$\egroup}

\DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \itshape
   \else
      \fontshape{ui}\selectfont
   \fi
   \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 LaTeX 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.

```
\DeclareTextComposite{\.}{OT1}{i}{`\i}
\DeclareTextComposite{\.}{OT1}{\i}{`\i}
\DeclareTextCompositeCommand{\`}{OT1}{i}{\@tabacckludge`\i}
\DeclareTextCompositeCommand{\'}{OT1}{i}{\@tabacckludge'\i}
\DeclareTextCompositeCommand{\^}{OT1}{i}{\^\i}
\DeclareTextCompositeCommand{\"}{OT1}{i}{\"\i}

⟨/OT1⟩
```

19.6 Definitions for the T1 encoding

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

Declare the encoding.

```
⟨*T1⟩
\DeclareFontEncoding{T1}{}{}
```

Declare the accents.

```
\DeclareTextAccent{\`}{T1}{0}
```

```
\DeclareTextAccent{\'}{T1}{1}
\DeclareTextAccent{\^}{T1}{2}
\DeclareTextAccent{\~}{T1}{3}
\DeclareTextAccent{\"}{T1}{4}
\DeclareTextAccent{\H}{T1}{5}
\DeclareTextAccent{\r}{T1}{6}
\DeclareTextAccent{\v}{T1}{7}
\DeclareTextAccent{\u}{T1}{8}
\DeclareTextAccent{\=}{T1}{9}
\DeclareTextAccent{\.}{T1}{10}
```

Some accents have to be built by hand. Note that `\ooalign` and `\o@lign` must be inside a group. In these definitions we no longer use the helper function `\sh@ft` from plain.tex since that now has two incompatible definitions.

```
\DeclareTextCommand{\b}{T1}[1]
   {\hmode@bgroup\o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-3ex}%
     \vbox to.2ex{\hbox{\char9}\vss}\hidewidth}\egroup}
\DeclareTextCommand{\c}{T1}[1]
   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent11 #1%
     \else{\ooalign{\unhbox\z@\crcr
        \hidewidth\char11\hidewidth}}\fi}
\DeclareTextCommand{\d}{T1}[1]
   {\hmode@bgroup
    \o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
\DeclareTextCommand{\k}{T1}[1]
   {\hmode@bgroup\ooalign{\null#1\crcr\hidewidth\char12}\egroup}
\DeclareTextCommand{\textogonekcentered}{T1}[1]
   {\hmode@bgroup\ooalign{\null#1\crcr\hidewidth\char12\hidewidth}\egroup}
```

Some symbols are constructed.

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

```
\DeclareTextCommand{\textperthousand}{T1}
   {\%\char 24 }          % space or `relax as delimiter?
\DeclareTextCommand{\textpertenthousand}{T1}
   {\%\char 24\char 24 }  % space or `relax as delimiter?
```

Declare the text symbols.

```
%\DeclareTextSymbol{\AA}{T1}{197}
\DeclareTextSymbol{\AE}{T1}{198}
\DeclareTextSymbol{\DH}{T1}{208}
\DeclareTextSymbol{\DJ}{T1}{208}
\DeclareTextSymbol{\L}{T1}{138}
\DeclareTextSymbol{\NG}{T1}{141}
\DeclareTextSymbol{\OE}{T1}{215}
\DeclareTextSymbol{\O}{T1}{216}
\DeclareTextSymbol{\SS}{T1}{223}
\DeclareTextSymbol{\TH}{T1}{222}
%\DeclareTextSymbol{\aa}{T1}{229}
\DeclareTextSymbol{\ae}{T1}{230}
\DeclareTextSymbol{\dh}{T1}{240}
\DeclareTextSymbol{\dj}{T1}{158}
\DeclareTextSymbol{\guillemotleft}{T1}{19}
\DeclareTextSymbol{\guillemotright}{T1}{20}
\DeclareTextSymbol{\guilsinglleft}{T1}{14}
\DeclareTextSymbol{\guilsinglright}{T1}{15}
\DeclareTextSymbol{\i}{T1}{25}
\DeclareTextSymbol{\j}{T1}{26}
\DeclareTextSymbol{\ij}{T1}{188}
\DeclareTextSymbol{\IJ}{T1}{156}
\DeclareTextSymbol{\l}{T1}{170}
\DeclareTextSymbol{\ng}{T1}{173}
\DeclareTextSymbol{\oe}{T1}{247}
\DeclareTextSymbol{\o}{T1}{248}
```

```
\DeclareTextSymbol{\quotedblbase}{T1}{18}
\DeclareTextSymbol{\quotesinglbase}{T1}{13}
\DeclareTextSymbol{\ss}{T1}{255}
\DeclareTextSymbol{\textasciicircum}{T1}{`\^}
\DeclareTextSymbol{\textasciitilde}{T1}{`\~}
\DeclareTextSymbol{\textbackslash}{T1}{`\\}
\DeclareTextSymbol{\textbar}{T1}{`\|}
\DeclareTextSymbol{\textbraceleft}{T1}{`\{}
\DeclareTextSymbol{\textbraceright}{T1}{`\}}
\DeclareTextSymbol{\textcompwordmark}{T1}{23}
\DeclareTextSymbol{\textdollar}{T1}{`\$}
\DeclareTextSymbol{\textemdash}{T1}{22}
\DeclareTextSymbol{\textendash}{T1}{21}
\DeclareTextSymbol{\textexclamdown}{T1}{189}
\DeclareTextSymbol{\textgreater}{T1}{`\>}
%\DeclareTextSymbol{\texthyphenchar}{T1}{127}
%\DeclareTextSymbol{\texthyphen}{T1}{`\-}
\DeclareTextSymbol{\textless}{T1}{`\<}
\DeclareTextSymbol{\textquestiondown}{T1}{190}
\DeclareTextSymbol{\textquotedblleft}{T1}{16}
\DeclareTextSymbol{\textquotedblright}{T1}{17}
\DeclareTextSymbol{\textquotedbl}{T1}{`\"}
\DeclareTextSymbol{\textquoteleft}{T1}{`\`}
\DeclareTextSymbol{\textquoteright}{T1}{`\'}
\DeclareTextSymbol{\textsection}{T1}{159}
\DeclareTextSymbol{\textsterling}{T1}{191}
\DeclareTextSymbol{\textunderscore}{T1}{95}
\DeclareTextSymbol{\textvisiblespace}{T1}{32}
\DeclareTextSymbol{\th}{T1}{254}
```

Declare the composites.

```
\DeclareTextComposite{\.}{T1}{i}{`\i}
\DeclareTextComposite{\.}{T1}{\i}{`\i}
```

”80 = 128

```
\DeclareTextComposite{\u}{T1}{A}{128}
\DeclareTextComposite{\k}{T1}{A}{129}
\DeclareTextComposite{\'}{T1}{C}{130}
\DeclareTextComposite{\v}{T1}{C}{131}
\DeclareTextComposite{\v}{T1}{D}{132}
\DeclareTextComposite{\v}{T1}{E}{133}
\DeclareTextComposite{\k}{T1}{E}{134}
\DeclareTextComposite{\u}{T1}{G}{135}
```

”88 = 136

```
\DeclareTextComposite{\'}{T1}{L}{136}
\DeclareTextComposite{\v}{T1}{L}{137}
\DeclareTextComposite{\'}{T1}{N}{139}
\DeclareTextComposite{\v}{T1}{N}{140}
\DeclareTextComposite{\H}{T1}{O}{142}
\DeclareTextComposite{\'}{T1}{R}{143}
```

”90 = 144

```
\DeclareTextComposite{\v}{T1}{R}{144}
\DeclareTextComposite{\'}{T1}{S}{145}
\DeclareTextComposite{\v}{T1}{S}{146}
\DeclareTextComposite{\c}{T1}{S}{147}
\DeclareTextComposite{\v}{T1}{T}{148}
\DeclareTextComposite{\c}{T1}{T}{149}
\DeclareTextComposite{\H}{T1}{U}{150}
\DeclareTextComposite{\r}{T1}{U}{151}
```

”98 = 152

```
\DeclareTextComposite{\"}{T1}{Y}{152}
```

```
\DeclareTextComposite{\'}{T1}{Z}{153}
\DeclareTextComposite{\v}{T1}{Z}{154}
\DeclareTextComposite{\.}{T1}{Z}{155}
\DeclareTextComposite{\.}{T1}{I}{157}
```

”A0 = 160

```
\DeclareTextComposite{\u}{T1}{a}{160}
\DeclareTextComposite{\k}{T1}{a}{161}
\DeclareTextComposite{\'}{T1}{c}{162}
\DeclareTextComposite{\v}{T1}{c}{163}
\DeclareTextComposite{\v}{T1}{d}{164}
\DeclareTextComposite{\v}{T1}{e}{165}
\DeclareTextComposite{\k}{T1}{e}{166}
\DeclareTextComposite{\u}{T1}{g}{167}
```

”A8 = 168

```
\DeclareTextComposite{\'}{T1}{l}{168}
\DeclareTextComposite{\v}{T1}{l}{169}
\DeclareTextComposite{\'}{T1}{n}{171}
\DeclareTextComposite{\v}{T1}{n}{172}
\DeclareTextComposite{\H}{T1}{o}{174}
\DeclareTextComposite{\'}{T1}{r}{175}
```

”B0 = 176

```
\DeclareTextComposite{\v}{T1}{r}{176}
\DeclareTextComposite{\'}{T1}{s}{177}
\DeclareTextComposite{\v}{T1}{s}{178}
\DeclareTextComposite{\c}{T1}{s}{179}
\DeclareTextComposite{\v}{T1}{t}{180}
\DeclareTextComposite{\c}{T1}{t}{181}
\DeclareTextComposite{\H}{T1}{u}{182}
\DeclareTextComposite{\r}{T1}{u}{183}
```

”B8 = 184

```
\DeclareTextComposite{\"}{T1}{y}{184}
\DeclareTextComposite{\'}{T1}{z}{185}
\DeclareTextComposite{\v}{T1}{z}{186}
\DeclareTextComposite{\.}{T1}{z}{187}
```

”C0 = 192

```
\DeclareTextComposite{\`}{T1}{A}{192}
\DeclareTextComposite{\'}{T1}{A}{193}
\DeclareTextComposite{\^}{T1}{A}{194}
\DeclareTextComposite{\~}{T1}{A}{195}
\DeclareTextComposite{\"}{T1}{A}{196}
\DeclareTextComposite{\r}{T1}{A}{197}
\DeclareTextComposite{\c}{T1}{C}{199}
```

”C8 = 200

```
\DeclareTextComposite{\`}{T1}{E}{200}
\DeclareTextComposite{\'}{T1}{E}{201}
\DeclareTextComposite{\^}{T1}{E}{202}
\DeclareTextComposite{\"}{T1}{E}{203}
\DeclareTextComposite{\`}{T1}{I}{204}
\DeclareTextComposite{\'}{T1}{I}{205}
\DeclareTextComposite{\^}{T1}{I}{206}
\DeclareTextComposite{\"}{T1}{I}{207}
```

”D0 = 208

```
\DeclareTextComposite{\~}{T1}{N}{209}
\DeclareTextComposite{\`}{T1}{O}{210}
\DeclareTextComposite{\'}{T1}{O}{211}
\DeclareTextComposite{\^}{T1}{O}{212}
\DeclareTextComposite{\~}{T1}{O}{213}
\DeclareTextComposite{\"}{T1}{O}{214}
```

”D8 = 216

```
\DeclareTextComposite{\`}{T1}{U}{217}
\DeclareTextComposite{\'}{T1}{U}{218}
\DeclareTextComposite{\^}{T1}{U}{219}
\DeclareTextComposite{\"}{T1}{U}{220}
\DeclareTextComposite{\'}{T1}{Y}{221}
```

”E0 = 224

```
\DeclareTextComposite{\`}{T1}{a}{224}
\DeclareTextComposite{\'}{T1}{a}{225}
\DeclareTextComposite{\^}{T1}{a}{226}
\DeclareTextComposite{\~}{T1}{a}{227}
\DeclareTextComposite{\"}{T1}{a}{228}
\DeclareTextComposite{\r}{T1}{a}{229}
\DeclareTextComposite{\c}{T1}{c}{231}
```

”E8 = 232

```
\DeclareTextComposite{\`}{T1}{e}{232}
\DeclareTextComposite{\'}{T1}{e}{233}
\DeclareTextComposite{\^}{T1}{e}{234}
\DeclareTextComposite{\"}{T1}{e}{235}
\DeclareTextComposite{\`}{T1}{i}{236}
\DeclareTextComposite{\`}{T1}{\i}{236}
\DeclareTextComposite{\'}{T1}{i}{237}
\DeclareTextComposite{\'}{T1}{\i}{237}
\DeclareTextComposite{\^}{T1}{i}{238}
\DeclareTextComposite{\^}{T1}{\i}{238}
\DeclareTextComposite{\"}{T1}{i}{239}
\DeclareTextComposite{\"}{T1}{\i}{239}
```

”F0 = 240

```
\DeclareTextComposite{\~}{T1}{n}{241}
\DeclareTextComposite{\`}{T1}{o}{242}
\DeclareTextComposite{\'}{T1}{o}{243}
\DeclareTextComposite{\^}{T1}{o}{244}
\DeclareTextComposite{\~}{T1}{o}{245}
\DeclareTextComposite{\"}{T1}{o}{246}
```

”F8 = 248

```
\DeclareTextComposite{\`}{T1}{u}{249}
\DeclareTextComposite{\'}{T1}{u}{250}
\DeclareTextComposite{\^}{T1}{u}{251}
\DeclareTextComposite{\"}{T1}{u}{252}
\DeclareTextComposite{\'}{T1}{y}{253}
```

```
\DeclareTextCompositeCommand{\k}{T1}{o}{\textogonekcentered{o}}
\DeclareTextCompositeCommand{\k}{T1}{O}{\textogonekcentered{O}}
```

```
⟨/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.

```
⟨*OMS⟩
\DeclareFontEncoding{OMS}{}{}
```

Declare the symbols.

```
% \changes{v1.99}{2004/02/02}{Added \cs{textbigcircle}}
%    Note that slot 13 has in places been named |\Orb|: please root
%    out and destroy this impolity wherever you find it!
%    \begin{macrocode}
\DeclareTextSymbol{\textasteriskcentered}{OMS}{3}   % "03
```

```
\DeclareTextSymbol{\textbackslash}{OMS}{110}      % "6E
\DeclareTextSymbol{\textbar}{OMS}{106}            % "6A
\DeclareTextSymbol{\textbardbl}{OMS}{107}         % "6B
\DeclareTextSymbol{\textbraceleft}{OMS}{102}      % "66
\DeclareTextSymbol{\textbraceright}{OMS}{103}     % "67
\DeclareTextSymbol{\textbullet}{OMS}{15}          % "0F
\DeclareTextSymbol{\textdaggerdbl}{OMS}{122}      % "7A
\DeclareTextSymbol{\textdagger}{OMS}{121}         % "79
\DeclareTextSymbol{\textparagraph}{OMS}{123}      % "7B
\DeclareTextSymbol{\textperiodcentered}{OMS}{1}   % "01
\DeclareTextSymbol{\textsection}{OMS}{120}        % "78
\DeclareTextSymbol{\textbigcircle}{OMS}{13}       % "0D
\DeclareTextCommand{\textcircled}{OMS}[1]{\hmode@bgroup
   \ooalign{%
      \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
      \char 13 % "0D
   }%
 \egroup}
⟨/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.

```
⟨*OML⟩
\DeclareFontEncoding{OML}{}{}
```

Declare the symbols.

```
\DeclareTextSymbol{\textless}{OML}{`\<}
\DeclareTextSymbol{\textgreater}{OML}{`\>}
\DeclareTextAccent{\t}{OML}{127}  % "7F
⟨/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.

```
⟨*OT4⟩
\DeclareFontEncoding{OT4}{}{}
\DeclareFontSubstitution{OT4}{cmr}{m}{n}
```

Declare the accents.

```
\DeclareTextAccent{\"}{OT4}{127}
\DeclareTextAccent{\'}{OT4}{19}
\DeclareTextAccent{\.}{OT4}{95}
\DeclareTextAccent{\=}{OT4}{22}
\DeclareTextAccent{\^}{OT4}{94}
\DeclareTextAccent{\`}{OT4}{18}
\DeclareTextAccent{\~}{OT4}{126}
\DeclareTextAccent{\H}{OT4}{125}
\DeclareTextAccent{\u}{OT4}{21}
\DeclareTextAccent{\v}{OT4}{20}
\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:

```
\DeclareTextCommand{\k}{OT4}[1]{%
    \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.

```
\DeclareTextCommand{\b}{OT4}[1]
   {\hmode@bgroup\o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-3ex}%
     \vbox to.2ex{\hbox{\char22}\vss}\hidewidth}\egroup}
\DeclareTextCommand{\c}{OT4}[1]
   {\leavevmode\setbox\z@\hbox{#1}\ifdim\ht\z@=1ex\accent24 #1%
    \else{\ooalign{\unhbox\z@\crcr\hidewidth\char24\hidewidth}}\fi}
\DeclareTextCommand{\d}{OT4}[1]
   {\hmode@bgroup
    \o@lign{\relax#1\crcr\hidewidth\ltx@sh@ft{-1ex}.\hidewidth}\egroup}
```

Declare the text symbols.

```
\DeclareTextSymbol{\AE}{OT4}{29}
\DeclareTextSymbol{\OE}{OT4}{30}
\DeclareTextSymbol{\O}{OT4}{31}
\DeclareTextSymbol{\L}{OT4}{138}
\DeclareTextSymbol{\ae}{OT4}{26}
\DeclareTextSymbol{\guillemotleft}{OT4}{174}
\DeclareTextSymbol{\guillemotright}{OT4}{175}
\DeclareTextSymbol{\i}{OT4}{16}
\DeclareTextSymbol{\j}{OT4}{17}
\DeclareTextSymbol{\l}{OT4}{170}
\DeclareTextSymbol{\o}{OT4}{28}
\DeclareTextSymbol{\oe}{OT4}{27}
\DeclareTextSymbol{\quotedblbase}{OT4}{255}
\DeclareTextSymbol{\ss}{OT4}{25}
\DeclareTextSymbol{\textemdash}{OT4}{124}
\DeclareTextSymbol{\textendash}{OT4}{123}
\DeclareTextSymbol{\textexclamdown}{OT4}{60}
%\DeclareTextSymbol{\texthyphenchar}{OT4}{`\-}
%\DeclareTextSymbol{\texthyphen}{OT4}{`\-}
\DeclareTextSymbol{\textquestiondown}{OT4}{62}
\DeclareTextSymbol{\textquotedblleft}{OT4}{92}
\DeclareTextSymbol{\textquotedblright}{OT4}{`\"}
\DeclareTextSymbol{\textquoteleft}{OT4}{`\`}
\DeclareTextSymbol{\textquoteright}{OT4}{`\'}
```

Definition for Å as in OT1:

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

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

```
\DeclareTextCommand{\textdollar}{OT4}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \slshape
   \else
      \upshape
   \fi
   \char`\$\egroup}
\DeclareTextCommand{\textsterling}{OT4}{\hmode@bgroup
   \ifdim \fontdimen\@ne\font >\z@
      \itshape
   \else
      \fontshape{ui}\selectfont
   \fi
   \char`\$\egroup}
```

Declare the composites.

```
\DeclareTextComposite{\k}{OT4}{A}{129}
\DeclareTextComposite{\'}{OT4}{C}{130}
\DeclareTextComposite{\k}{OT4}{E}{134}
\DeclareTextComposite{\'}{OT4}{N}{139}
\DeclareTextComposite{\'}{OT4}{S}{145}
\DeclareTextComposite{\'}{OT4}{Z}{153}
\DeclareTextComposite{\.}{OT4}{Z}{155}
\DeclareTextComposite{\k}{OT4}{a}{161}
\DeclareTextComposite{\'}{OT4}{c}{162}
\DeclareTextComposite{\k}{OT4}{e}{166}
\DeclareTextComposite{\'}{OT4}{n}{171}
\DeclareTextComposite{\'}{OT4}{s}{177}
\DeclareTextComposite{\'}{OT4}{z}{185}
\DeclareTextComposite{\.}{OT4}{z}{187}
\DeclareTextComposite{\'}{OT4}{O}{211}
\DeclareTextComposite{\'}{OT4}{o}{243}
⟨/OT4⟩
```

19.10 Definitions for the TS1 encoding

```
⟨*TS1⟩
\DeclareFontEncoding{TS1}{}{}
\DeclareFontSubstitution{TS1}{cmr}{m}{n}
```

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

```
\DeclareTextCommand{\capitalcedilla}{TS1}[1]
   {\hmode@bgroup
    \ooalign{\null#1\crcr\hidewidth\char11\hidewidth}\egroup}
\DeclareTextCommand{\capitalogonek}{TS1}[1]
   {\hmode@bgroup
    \ooalign{\null#1\crcr\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

```
\DeclareTextAccent{\capitalgrave}{TS1}{0}
\DeclareTextAccent{\capitalacute}{TS1}{1}
\DeclareTextAccent{\capitalcircumflex}{TS1}{2}
\DeclareTextAccent{\capitaltilde}{TS1}{3}
\DeclareTextAccent{\capitaldieresis}{TS1}{4}
\DeclareTextAccent{\capitalhungarumlaut}{TS1}{5}
\DeclareTextAccent{\capitalring}{TS1}{6}
\DeclareTextAccent{\capitalcaron}{TS1}{7}
```

”08 = 8

```
\DeclareTextAccent{\capitalbreve}{TS1}{8}
\DeclareTextAccent{\capitalmacron}{TS1}{9}
\DeclareTextAccent{\capitaldotaccent}{TS1}{10}
```

Tie accents.

The tie accent was borrowed from the `cmmi` font. The tc fonts now provide four tie accents, the first two are done in the classical way with assymetric glyphs hanging out of their boxes; the new ties are centered in their boxes like all other accents. They need a name: please tell us if you know what to call them.

” =

```
\DeclareTextAccent{\t}{TS1}{26}
\DeclareTextAccent{\capitaltie}{TS1}{27}
\DeclareTextAccent{\newtie}{TS1}{28}
\DeclareTextAccent{\capitalnewtie}{TS1}{29}
```

Compund word marks.

The text companion fonts contain two compound word marks of different heights, one has `cap_height`, the other `asc_height`.

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

The text companion symbols.

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

”10 = 16

```
\DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
\DeclareTextSymbol{\texttwelveudash}{TS1}{21}
\DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
```

”18 = 24

```
\DeclareTextSymbol{\textleftarrow}{TS1}{24}
\DeclareTextSymbol{\textrightarrow}{TS1}{25}
```

”20 = 32

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

”28 = 40

```
\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.

```
\DeclareTextSymbol{\textdblhyphen}{TS1}{45}
\DeclareTextSymbol{\textfractionsolidus}{TS1}{47}
```

Oldstyle digits.

”30 = 48

```
\DeclareTextSymbol{\textzerooldstyle}{TS1}{48}
\DeclareTextSymbol{\textoneoldstyle}{TS1}{49}
\DeclareTextSymbol{\texttwooldstyle}{TS1}{50}
\DeclareTextSymbol{\textthreeoldstyle}{TS1}{51}
\DeclareTextSymbol{\textfouroldstyle}{TS1}{52}
\DeclareTextSymbol{\textfiveoldstyle}{TS1}{53}
\DeclareTextSymbol{\textsixoldstyle}{TS1}{54}
\DeclareTextSymbol{\textsevenoldstyle}{TS1}{55}
```

”38 = 56

```
\DeclareTextSymbol{\texteightoldstyle}{TS1}{56}
\DeclareTextSymbol{\textnineoldstyle}{TS1}{57}
```

More text companion symbols.

```
\DeclareTextSymbol{\textlangle}{TS1}{60}
\DeclareTextSymbol{\textminus}{TS1}{61}
\DeclareTextSymbol{\textrangle}{TS1}{62}
```

”48 = 72

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

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

```
\DeclareTextSymbol{\textbigcircle}{TS1}{79}
\DeclareTextCommand{\textcircled}{TS1}[1]{\hmode@bgroup
  \ooalign{%
     \hfil \raise .07ex\hbox {\upshape#1}\hfil \crcr
     \char 79   % '117 = "4F
  }%
\egroup}
```

More text companion symbols.

”50 = 80

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

”58 = 88

```
\DeclareTextSymbol{\textlbrackdbl}{TS1}{91}
\DeclareTextSymbol{\textrbrackdbl}{TS1}{93}
\DeclareTextSymbol{\textuparrow}{TS1}{94}
\DeclareTextSymbol{\textdownarrow}{TS1}{95}
```

”60 = 96

```
\DeclareTextSymbol{\textasciigrave}{TS1}{96}
\DeclareTextSymbol{\textborn}{TS1}{98}
\DeclareTextSymbol{\textdivorced}{TS1}{99}
\DeclareTextSymbol{\textdied}{TS1}{100}
```

”68 = 104

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

”78 = 120

```
\DeclareTextSymbol{\texttildelow}{TS1}{126}
```

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

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

”80 = 128

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

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

```
\DeclareTextSymbol{\textacutedbl}{TS1}{130}
\DeclareTextSymbol{\textgravedbl}{TS1}{131}
\DeclareTextSymbol{\textdagger}{TS1}{132}
\DeclareTextSymbol{\textdaggerdbl}{TS1}{133}
\DeclareTextSymbol{\textbardbl}{TS1}{134}
\DeclareTextSymbol{\textperthousand}{TS1}{135}
```

”88 = 136

```
\DeclareTextSymbol{\textbullet}{TS1}{136}
\DeclareTextSymbol{\textcelsius}{TS1}{137}
\DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
\DeclareTextSymbol{\textcentoldstyle}{TS1}{139}
\DeclareTextSymbol{\textflorin}{TS1}{140}
\DeclareTextSymbol{\textcolonmonetary}{TS1}{141}
\DeclareTextSymbol{\textwon}{TS1}{142}
\DeclareTextSymbol{\textnaira}{TS1}{143}
```

”90 = 144

```
\DeclareTextSymbol{\textguarani}{TS1}{144}
\DeclareTextSymbol{\textpeso}{TS1}{145}
\DeclareTextSymbol{\textlira}{TS1}{146}
\DeclareTextSymbol{\textrecipe}{TS1}{147}
\DeclareTextSymbol{\textinterrobang}{TS1}{148}
\DeclareTextSymbol{\textinterrobangdown}{TS1}{149}
\DeclareTextSymbol{\textdong}{TS1}{150}
\DeclareTextSymbol{\texttrademark}{TS1}{151}
```

”98 = 152

```
\DeclareTextSymbol{\textpertenthousand}{TS1}{152}
\DeclareTextSymbol{\textpilcrow}{TS1}{153}
\DeclareTextSymbol{\textbaht}{TS1}{154}
\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`.

```
\DeclareTextSymbol{\textdiscount}{TS1}{156}
\DeclareTextSymbol{\textestimated}{TS1}{157}
\DeclareTextSymbol{\textopenbullet}{TS1}{158}
\DeclareTextSymbol{\textservicemark}{TS1}{159}
```

”A0 = 160

```
\DeclareTextSymbol{\textlquill}{TS1}{160}
\DeclareTextSymbol{\textrquill}{TS1}{161}
\DeclareTextSymbol{\textcent}{TS1}{162}
\DeclareTextSymbol{\textsterling}{TS1}{163}
\DeclareTextSymbol{\textcurrency}{TS1}{164}
\DeclareTextSymbol{\textyen}{TS1}{165}
\DeclareTextSymbol{\textbrokenbar}{TS1}{166}
\DeclareTextSymbol{\textsection}{TS1}{167}
```

”A8 = 168

```
\DeclareTextSymbol{\textasciidieresis}{TS1}{168}
\DeclareTextSymbol{\textcopyright}{TS1}{169}
\DeclareTextSymbol{\textordfeminine}{TS1}{170}
\DeclareTextSymbol{\textcopyleft}{TS1}{171}
\DeclareTextSymbol{\textlnot}{TS1}{172}
```

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

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

”B0 = 176

```
\DeclareTextSymbol{\textdegree}{TS1}{176}
\DeclareTextSymbol{\textpm}{TS1}{177}
\DeclareTextSymbol{\texttwosuperior}{TS1}{178}
\DeclareTextSymbol{\textthreesuperior}{TS1}{179}
\DeclareTextSymbol{\textasciiacute}{TS1}{180}
\DeclareTextSymbol{\textmu}{TS1}{181} % micro sign
\DeclareTextSymbol{\textparagraph}{TS1}{182}
\DeclareTextSymbol{\textperiodcentered}{TS1}{183}
```

”B8 = 184

```
\DeclareTextSymbol{\textreferencemark}{TS1}{184}
\DeclareTextSymbol{\textonesuperior}{TS1}{185}
\DeclareTextSymbol{\textordmasculine}{TS1}{186}
\DeclareTextSymbol{\textsurd}{TS1}{187}
\DeclareTextSymbol{\textonequarter}{TS1}{188}
\DeclareTextSymbol{\textonehalf}{TS1}{189}
\DeclareTextSymbol{\textthreequarters}{TS1}{190}
\DeclareTextSymbol{\texteuro}{TS1}{191}
```

”E0 = 208

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

”F0 = 240

```
\DeclareTextSymbol{\textdiv}{TS1}{246}
⟨/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 `FOO`, the package looks to see if the encoding `FOO` has already been declared. If it has not, the file `fooenc.def` is loaded. The default encoding is set to be `FOO`.

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

⟨*package⟩

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

```
\def\update@uclc@with@cyrillic{%
 \expandafter\def\expandafter\@uclclist\expandafter
  {\@uclclist
  \cyra\CYRA\cyrabhch\CYRABHCH\cyrabhchdsc\CYRABHCHDSC\cyrabhdze
  \CYRABHDZE\cyrabhha\CYRABHHA\cyrae\CYRAE\cyrb\CYRB\cyrbyus
  \CYRBYUS\cyrc\CYRC\cyrch\CYRCH\cyrchldsc\CYRCHLDSC\cyrchrdsc
  \CYRCHRDSC\cyrchvcrs\CYRCHVCRS\cyrd\CYRD\cyrdelta\CYRDELTA
  \cyrdje\CYRDJE\cyrdze\CYRDZE\cyrdzhe\CYRDZHE\cyre\CYRE\cyreps
  \CYREPS\cyrerev\CYREREV\cyrery\CYRERY\cyrf\CYRF\cyrfita
  \CYRFITA\cyrg\CYRG\cyrgdsc\CYRGDSC\cyrgdschcrs\CYRGDSCHCRS
  \cyrghcrs\CYRGHCRS\cyrghk\CYRGHK\cyrgup\CYRGUP\cyrh\CYRH
  \cyrhdsc\CYRHDSC\cyrhhcrs\CYRHHCRS\cyrhhk\CYRHHK\cyrhrdsn
  \CYRHRDSN\cyri\CYRI\cyrie\CYRIE\cyrii\CYRII\cyrishrt\CYRISHRT
  \cyrishrtdsc\CYRISHRTDSC\cyrizh\CYRIZH\cyrje\CYRJE\cyrk\CYRK
  \cyrkbeak\CYRKBEAK\cyrkdsc\CYRKDSC\cyrkhcrs\CYRKHCRS\cyrkhk
  \CYRKHK\cyrkvcrs\CYRKVCRS\cyrl\CYRL\cyrldsc\CYRLDSC\cyrlhk
  \CYRLHK\cyrlje\CYRLJE\cyrm\CYRM\cyrmdsc\CYRMDSC\cyrmhk\CYRMHK
  \cyrn\CYRN\cyrndsc\CYRNDSC\cyrng\CYRNG\cyrnhk\CYRNHK\cyrnje
  \CYRNJE\cyrnlhk\CYRNLHK\cyro\CYRO\cyrotld\CYROTLD\cyrp\CYRP
  \cyrphk\CYRPHK\cyrq\CYRQ\cyrr\CYRR\cyrrdsc\CYRRDSC\cyrrhk
  \CYRRHK\cyrrtick\CYRRTICK\cyrs\CYRS\cyrsacrs\CYRSACRS
  \cyrschwa\CYRSCHWA\cyrsdsc\CYRSDSC\cyrsemisftsn\CYRSEMISFTSN
  \cyrsftsn\CYRSFTSN\cyrsh\CYRSH\cyrshch\CYRSHCH\cyrshha\CYRSHHA
  \cyrt\CYRT\cyrtdsc\CYRTDSC\cyrtetse\CYRTETSE\cyrtshe\CYRTSHE
  \cyru\CYRU\cyrushrt\CYRUSHRT\cyrv\CYRV\cyrw\CYRW\cyry\CYRY
  \cyrya\CYRYA\cyryat\CYRYAT\cyryhcrs\CYRYHCRS\cyryi\CYRYI\cyryo
  \CYRYO\cyryu\CYRYU\cyrz\CYRZ\cyrzdsc\CYRZDSC\cyrzh\CYRZH
  \cyrzhdsc\CYRZHDSC}%
 \let\update@uclc@with@cyrillic\relax
}
```

Here we process each option:

```
\DeclareOption*{%
   \let\encodingdefault\CurrentOption
   \edef\reserved@f{%
     \lowercase{\def\noexpand\reserved@f{\CurrentOption enc.def}}}%
   \reserved@f
   \InputIfFileExists\reserved@f
        {}{\PackageError{fontenc}%
         {Encoding file `\reserved@f' not found.%
          \MessageBreak
           You might have misspelt the name of the encoding}%
         {Necessary code for this encoding was not
          loaded.\MessageBreak
          Thus calling the encoding later on will
          produce further error messages.}}%
  \let\reserved@f\relax
```

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

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

```
    \expandafter\in@\expandafter\cyra\expandafter
                             {\@uclclist}%
    \ifin@
    \else
      \update@uclc@with@cyrillic
    \fi
  \fi
}
```

```
\ProcessOptions*
```

```
\fontencoding\encodingdefault\selectfont
```

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

```
\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.

```
\global\expandafter\let\csname ver@fontenc.sty\endcsname\relax
\global\expandafter\let\csname opt@fontenc.sty\endcsname\relax
\global\let\@ifl@ter@@\@ifl@ter
\def\@ifl@ter#1#2#3#4#5{\global\let\@ifl@ter\@ifl@ter@@}
⟨/package⟩
```

20.2 The textcomp package

This one is for the `TS1` encoding which contains text symbols for use with the `T1`-encoded text fonts. It therefore first inputs the file `TS1enc.def` and then sets (or resets) the defaults for the symbols it contains. The result of this is that when one of these symbols is accessed and the current encoding does not provide it, the symbol will be supplied by a silent, local change to this encoding.

```
⟨*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`∩(`ISO-Adobe`∪`MacRoman`). (Except for the missing "currency".)

#2 = #3 + `\textestimated` + `\textcurrency`. Can also be described as `TS1` ∩ `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:

```
\PackageInfo{textcomp}{Sub-encoding information:\MessageBreak
    \space\space 5 = only ISO-Adobe without \string\textcurrency\MessageBreak
    \space\space 4 = 5 + \string\texteuro\MessageBreak
    \space\space 3 = 4 + \string\textohm\MessageBreak
    \space\space 2 = 3 + \noexpand\textestimated+ \string\textcurrency\MessageBreak
    \space\space 1 = TS1 - \noexpand\textcircled- \string\t\MessageBreak
    \space\space 0 = TS1 (full)\MessageBreak
    Font families with sub-encoding setting implement\MessageBreak
    only a restricted character set as indicated.\MessageBreak
    Family '?' is the default used for unknown fonts.\MessageBreak
    See the documentation for details\@gobble}
```

`\DeclareEncodingSubset` An encoding subset to which a font family belongs is declared by `\DeclareEncodingSubset` that take 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 `?`.

```
\def\DeclareEncodingSubset#1#2#3{%
   \@ifundefined{#1:#2}%
     {\PackageInfo{textcomp}{Setting #2 sub-encoding to #1/#3}}%
     {\PackageInfo{textcomp}{Changing #2 sub-encoding to #1/#3}}%
   \@namedef{#1:#2}{#3}}
\@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 specifed by one of the other options (seldom useful, only dangerous).

`\iftc@forced` Switch used to implement the `force` option

```
\newif\iftc@forced   \tc@forcedfalse
```

This is implemented by defining the default subset:

```
\DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
\DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
\DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{4}}
\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.

```
\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.

```
\def\tc@errorwarn{\PackageError}
\DeclareOption{warn}{\gdef\tc@errorwarn#1#2#3{\PackageWarning{#1}{#2}}}
```

```
\ExecuteOptions{almostfull}
\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.

```
\iftc@forced
```

If the “force” option was given we always use the default for testing against.

```
\def\CheckEncodingSubset#1#2#3#4#5{%
    \ifnum #4>%
        0\csname #2:?\endcsname
        \relax
   \expandafter\@firstoftwo
  \else
   \expandafter\@secondoftwo
 \fi
  {#1{#2}}{#3}%
  #5%
}
```

In normal circumstances the test is a bit more complicated: first check if there exists a macro \⟨*arg2*⟩`:`⟨*current-family*⟩ and if so use that value to test against, otherwise use the default to test against.

```
\else
\def\CheckEncodingSubset#1#2#3#4#5{%
    \ifnum #4>%
      \expandafter\ifx\csname #2:\f@family\endcsname\relax
        0\csname #2:?\endcsname
      \else
        \csname #2:\f@family\endcsname
      \fi
   \relax
   \expandafter\@firstoftwo
  \else
   \expandafter\@secondoftwo
 \fi
  {#1{#2}}{#3}%
  #5%
}
\fi
```

`tc@subst`

```
\def\tc@subst#1{%
   \tc@errorwarn{textcomp}%  % should be latex error if general
    {Symbol \string#1 not provided by\MessageBreak
     font family \f@family\space
     in TS1 encoding.\MessageBreak Default family used instead}\@eha
  \bgroup\fontfamily\textcompsubstdefault\selectfont#1\egroup
}
```

`\textcompsubstdefault`

```
\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.

```
% error commands take argument:
% #1 symbol to be used
\def\tc@error#1{%
   \PackageError{textcomp}%  % should be latex error if general
    {Accent \string#1 not provided by\MessageBreak
     font family \f@family\space
     in TS1 encoding}\@eha
}
```

`\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 “=”.

```
\def\tc@fake@euro#1{%
   \leavevmode
   \PackageInfo{textcomp}{Faking \noexpand#1for font family
                         \f@family\MessageBreak in TS1 encoding}%
   \valign{##\cr
      \vfil\hbox to 0.07em{\dimen@\f@size\p@
                           \math@fontsfalse
                           \fontsize{.7\dimen@}\z@\selectfont=\hss}\vfil\cr%
      \hbox{C}\crcr
   }%
}
```

`\tc@check@symbol`
`\tc@check@accent`
These are two abbreviations that we use below to check symbols and accents in 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 `\textcurreny` is not available.

```
\def\tc@check@symbol{\CheckEncodingSubset\UseTextSymbol{TS1}\tc@subst}
\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. . .

```
\DeclareTextAccentDefault{\capitalcedilla}{TS1}
\DeclareTextAccentDefault{\capitalogonek}{TS1}
\DeclareTextAccentDefault{\capitalgrave}{TS1}
\DeclareTextAccentDefault{\capitalacute}{TS1}
\DeclareTextAccentDefault{\capitalcircumflex}{TS1}
\DeclareTextAccentDefault{\capitaltilde}{TS1}
```

```
\DeclareTextAccentDefault{\capitaldieresis}{TS1}
\DeclareTextAccentDefault{\capitalhungarumlaut}{TS1}
\DeclareTextAccentDefault{\capitalring}{TS1}
\DeclareTextAccentDefault{\capitalcaron}{TS1}
\DeclareTextAccentDefault{\capitalbreve}{TS1}
\DeclareTextAccentDefault{\capitalmacron}{TS1}
\DeclareTextAccentDefault{\capitaldotaccent}{TS1}
```

. . . and then the other glyphs.

```
\DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
\DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
\DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
\DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
\DeclareTextSymbolDefault{\texttwelveudash}{TS1}
\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1}
\DeclareTextSymbolDefault{\textdollar}{TS1}
\DeclareTextSymbolDefault{\textquotesingle}{TS1}
\DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
\DeclareTextSymbolDefault{\textfractionsolidus}{TS1}
\DeclareTextSymbolDefault{\textminus}{TS1}
\DeclareTextSymbolDefault{\textlbrackdbl}{TS1}
\DeclareTextSymbolDefault{\textrbrackdbl}{TS1}
\DeclareTextSymbolDefault{\textasciigrave}{TS1}
\DeclareTextSymbolDefault{\texttildelow}{TS1}
\DeclareTextSymbolDefault{\textasciibreve}{TS1}
\DeclareTextSymbolDefault{\textasciicaron}{TS1}
\DeclareTextSymbolDefault{\textgravedbl}{TS1}
\DeclareTextSymbolDefault{\textacutedbl}{TS1}
\DeclareTextSymbolDefault{\textdagger}{TS1}
\DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
\DeclareTextSymbolDefault{\textbardbl}{TS1}
\DeclareTextSymbolDefault{\textperthousand}{TS1}
\DeclareTextSymbolDefault{\textbullet}{TS1}
\DeclareTextSymbolDefault{\textcelsius}{TS1}
\DeclareTextSymbolDefault{\textflorin}{TS1}
\DeclareTextSymbolDefault{\texttrademark}{TS1}
\DeclareTextSymbolDefault{\textcent}{TS1}
\DeclareTextSymbolDefault{\textsterling}{TS1}
\DeclareTextSymbolDefault{\textyen}{TS1}
\DeclareTextSymbolDefault{\textbrokenbar}{TS1}
\DeclareTextSymbolDefault{\textsection}{TS1}
\DeclareTextSymbolDefault{\textasciidieresis}{TS1}
\DeclareTextSymbolDefault{\textcopyright}{TS1}
\DeclareTextSymbolDefault{\textordfeminine}{TS1}
\DeclareTextSymbolDefault{\textlnot}{TS1}
\DeclareTextSymbolDefault{\textregistered}{TS1}
\DeclareTextSymbolDefault{\textasciimacron}{TS1}
\DeclareTextSymbolDefault{\textdegree}{TS1}
\DeclareTextSymbolDefault{\textpm}{TS1}
\DeclareTextSymbolDefault{\texttwosuperior}{TS1}
\DeclareTextSymbolDefault{\textthreesuperior}{TS1}
\DeclareTextSymbolDefault{\textasciiacute}{TS1}
\DeclareTextSymbolDefault{\textmu}{TS1}
\DeclareTextSymbolDefault{\textparagraph}{TS1}
\DeclareTextSymbolDefault{\textperiodcentered}{TS1}
\DeclareTextSymbolDefault{\textonesuperior}{TS1}
\DeclareTextSymbolDefault{\textordmasculine}{TS1}
\DeclareTextSymbolDefault{\textonequarter}{TS1}
\DeclareTextSymbolDefault{\textonehalf}{TS1}
\DeclareTextSymbolDefault{\textthreequarters}{TS1}
\DeclareTextSymbolDefault{\texttimes}{TS1}
\DeclareTextSymbolDefault{\textdiv}{TS1}
```

The `\texteuro` is only available for subsets with id 4 or less. Otherwise we fake the glyph using `\tc@fake@euro`

```
\DeclareTextCommandDefault{\texteuro}
   {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro5\texteuro}
```

The `\textohm` is only available for subsets with id 3 or less. Otherwise we produce an error.

```
\DeclareTextCommandDefault{\textohm}{\tc@check@symbol4\textohm}
```

The `\textestimated` and `\textcurrency` are only provided for fonts with subset encoding with id 2 or less.

```
\DeclareTextCommandDefault{\textestimated}{\tc@check@symbol3\textestimated}
\DeclareTextCommandDefault{\textcurrency}{\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.

```
\DeclareTextCommandDefault{\capitaltie}{\tc@check@accent2\capitaltie}
\DeclareTextCommandDefault{\newtie}{\tc@check@accent2\newtie}
\DeclareTextCommandDefault{\capitalnewtie}{\tc@check@accent2\capitalnewtie}
\DeclareTextCommandDefault{\textleftarrow}{\tc@check@symbol2\textleftarrow}
\DeclareTextCommandDefault{\textrightarrow}{\tc@check@symbol2\textrightarrow}
\DeclareTextCommandDefault{\textblank}{\tc@check@symbol2\textblank}
\DeclareTextCommandDefault{\textdblhyphen}{\tc@check@symbol2\textdblhyphen}
\DeclareTextCommandDefault{\textzerooldstyle}{\tc@check@symbol2\textzerooldstyle}
\DeclareTextCommandDefault{\textoneoldstyle}{\tc@check@symbol2\textoneoldstyle}
\DeclareTextCommandDefault{\texttwooldstyle}{\tc@check@symbol2\texttwooldstyle}
\DeclareTextCommandDefault{\textthreeoldstyle}{\tc@check@symbol2\textthreeoldstyle}
\DeclareTextCommandDefault{\textfouroldstyle}{\tc@check@symbol2\textfouroldstyle}
\DeclareTextCommandDefault{\textfiveoldstyle}{\tc@check@symbol2\textfiveoldstyle}
\DeclareTextCommandDefault{\textsixoldstyle}{\tc@check@symbol2\textsixoldstyle}
\DeclareTextCommandDefault{\textsevenoldstyle}{\tc@check@symbol2\textsevenoldstyle}
\DeclareTextCommandDefault{\texteightoldstyle}{\tc@check@symbol2\texteightoldstyle}
\DeclareTextCommandDefault{\textnineoldstyle}{\tc@check@symbol2\textnineoldstyle}
\DeclareTextCommandDefault{\textlangle}{\tc@check@symbol2\textlangle}
\DeclareTextCommandDefault{\textrangle}{\tc@check@symbol2\textrangle}
\DeclareTextCommandDefault{\textmho}{\tc@check@symbol2\textmho}
\DeclareTextCommandDefault{\textbigcircle}{\tc@check@symbol2\textbigcircle}
\DeclareTextCommandDefault{\textuparrow}{\tc@check@symbol2\textuparrow}
\DeclareTextCommandDefault{\textdownarrow}{\tc@check@symbol2\textdownarrow}
\DeclareTextCommandDefault{\textborn}{\tc@check@symbol2\textborn}
\DeclareTextCommandDefault{\textdivorced}{\tc@check@symbol2\textdivorced}
\DeclareTextCommandDefault{\textdied}{\tc@check@symbol2\textdied}
\DeclareTextCommandDefault{\textleaf}{\tc@check@symbol2\textleaf}
\DeclareTextCommandDefault{\textmarried}{\tc@check@symbol2\textmarried}
\DeclareTextCommandDefault{\textmusicalnote}{\tc@check@symbol2\textmusicalnote}
\DeclareTextCommandDefault{\textdblhyphenchar}{\tc@check@symbol2\textdblhyphenchar}
\DeclareTextCommandDefault{\textdollaroldstyle}{\tc@check@symbol2\textdollaroldstyle}
\DeclareTextCommandDefault{\textcentoldstyle}{\tc@check@symbol2\textcentoldstyle}
\DeclareTextCommandDefault{\textcolonmonetary}{\tc@check@symbol2\textcolonmonetary}
\DeclareTextCommandDefault{\textwon}{\tc@check@symbol2\textwon}
\DeclareTextCommandDefault{\textnaira}{\tc@check@symbol2\textnaira}
\DeclareTextCommandDefault{\textguarani}{\tc@check@symbol2\textguarani}
\DeclareTextCommandDefault{\textpeso}{\tc@check@symbol2\textpeso}
\DeclareTextCommandDefault{\textlira}{\tc@check@symbol2\textlira}
\DeclareTextCommandDefault{\textrecipe}{\tc@check@symbol2\textrecipe}
\DeclareTextCommandDefault{\textinterrobang}{\tc@check@symbol2\textinterrobang}
\DeclareTextCommandDefault{\textinterrobangdown}{\tc@check@symbol2\textinterrobangdown}
\DeclareTextCommandDefault{\textdong}{\tc@check@symbol2\textdong}
\DeclareTextCommandDefault{\textpertenthousand}{\tc@check@symbol2\textpertenthousand}
\DeclareTextCommandDefault{\textpilcrow}{\tc@check@symbol2\textpilcrow}
\DeclareTextCommandDefault{\textbaht}{\tc@check@symbol2\textbaht}
\DeclareTextCommandDefault{\textnumero}{\tc@check@symbol2\textnumero}
\DeclareTextCommandDefault{\textdiscount}{\tc@check@symbol2\textdiscount}
```

```
\DeclareTextCommandDefault{\textopenbullet}{\tc@check@symbol2\textopenbullet}
\DeclareTextCommandDefault{\textservicemark}{\tc@check@symbol2\textservicemark}
\DeclareTextCommandDefault{\textlquill}{\tc@check@symbol2\textlquill}
\DeclareTextCommandDefault{\textrquill}{\tc@check@symbol2\textrquill}
\DeclareTextCommandDefault{\textcopyleft}{\tc@check@symbol2\textcopyleft}
\DeclareTextCommandDefault{\textcircledP}{\tc@check@symbol2\textcircledP}
\DeclareTextCommandDefault{\textreferencemark}{\tc@check@symbol2\textreferencemark}
\DeclareTextCommandDefault{\textsurd}{\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.

```
\DeclareTextCommandDefault{\textcircled}
  {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OMS}}1\textcircled}
\DeclareTextCommandDefault{\t}
  {\CheckEncodingSubset\UseTextAccent{TS1}{\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).

```
\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 LaTeX which will prevent the usage of the `TS1` versions (see section 19.1 above). So we better get rid of them:

```
\UndeclareTextCommand{\textsterling}{OT1}
\UndeclareTextCommand{\textdollar}  {OT1}
```

Similar declarations should probably be made for other encodings like `OT4` if they are in use.

```
%\UndeclareTextCommand{\textsterling}{OT4}
%\UndeclareTextCommand{\textdollar}  {OT4}
```

From the `T1` encoding there are two candidates for removal: ‰ and ‱ since these are both constructed from % followed by a tiny '₀' 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 ‰ 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 ‰ and ‱ are not taken from the same physical font) and with PostScript fonts ‰ will come out correctly while ‱ 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`.

```
\UndeclareTextCommand{\textperthousand}{T1}
%\UndeclareTextCommand{\textpertenthousand}{T1}
```

20.2.1 Supporting oldstyle digits

```
\DeclareRobustCommand\oldstylenums[1]{%
 \begingroup
  \ifmmode
   \mathgroup\symletters #1%
  \else
   \CheckEncodingSubset\@use@text@encoding{TS1}%
       {\PackageWarning{textcomp}%
          {Oldstyle digits unavailable for
           family \f@family.\MessageBreak
           Lining digits used instead}}%
       \tw@{#1}%
   \fi
 \endgroup
}
```

20.2.2 Subset encoding defaults

For many font families commonly used in the TEX 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:

```
\iftc@forced \else
```

Computer modern based fonts (e.g., CM, CM-Bright, Concrete):

```
\DeclareEncodingSubset{TS1}{cmr}     {0}
\DeclareEncodingSubset{TS1}{cmss}    {0}
\DeclareEncodingSubset{TS1}{cmtt}    {0}
\DeclareEncodingSubset{TS1}{cmvtt}   {0}
\DeclareEncodingSubset{TS1}{cmbr}    {0}
\DeclareEncodingSubset{TS1}{cmtl}    {0}
\DeclareEncodingSubset{TS1}{ccr}     {0}
```

PSNFSS fonts:

```
\DeclareEncodingSubset{TS1}{ptm}     {4}
\DeclareEncodingSubset{TS1}{pcr}     {4}
\DeclareEncodingSubset{TS1}{phv}     {4}
\DeclareEncodingSubset{TS1}{ppl}     {3}
\DeclareEncodingSubset{TS1}{pag}     {4}
\DeclareEncodingSubset{TS1}{pbk}     {4}
\DeclareEncodingSubset{TS1}{pnc}     {4}
\DeclareEncodingSubset{TS1}{pzc}     {4}
\DeclareEncodingSubset{TS1}{bch}     {4}
\DeclareEncodingSubset{TS1}{put}     {5}
```

Other CTAN fonts (probably not complete):

```
\DeclareEncodingSubset{TS1}{uag}     {5}
\DeclareEncodingSubset{TS1}{ugq}     {5}
\DeclareEncodingSubset{TS1}{ul8}     {4}
\DeclareEncodingSubset{TS1}{ul9}     {4}  % (LuxiSans, one day)
\DeclareEncodingSubset{TS1}{augie}   {5}
\DeclareEncodingSubset{TS1}{dayrom}  {3}
\DeclareEncodingSubset{TS1}{dayroms} {3}
\DeclareEncodingSubset{TS1}{pxr}     {0}
\DeclareEncodingSubset{TS1}{pxss}    {0}
\DeclareEncodingSubset{TS1}{pxtt}    {0}
\DeclareEncodingSubset{TS1}{txr}     {0}
\DeclareEncodingSubset{TS1}{txss}    {0}
\DeclareEncodingSubset{TS1}{txtt}    {0}
```

Latin Modern and TeX Gyre:

```
\DeclareEncodingSubset{TS1}{lmr}     {0}
\DeclareEncodingSubset{TS1}{lmdh}    {0}
\DeclareEncodingSubset{TS1}{lmss}    {0}
\DeclareEncodingSubset{TS1}{lmssq}   {0}
\DeclareEncodingSubset{TS1}{lmvtt}   {0}
```

```
\DeclareEncodingSubset{TS1}{qhv}     {0}
\DeclareEncodingSubset{TS1}{qag}     {0}
\DeclareEncodingSubset{TS1}{qbk}     {0}
\DeclareEncodingSubset{TS1}{qcr}     {0}
\DeclareEncodingSubset{TS1}{qcs}     {0}
\DeclareEncodingSubset{TS1}{qpl}     {0}
\DeclareEncodingSubset{TS1}{qtm}     {0}
\DeclareEncodingSubset{TS1}{qzc}     {0}
\DeclareEncodingSubset{TS1}{qhvc}    {0}
```

Fourier-GUTenberg:

```
\DeclareEncodingSubset{TS1}{futs}    {4}
\DeclareEncodingSubset{TS1}{futx}    {4}
\DeclareEncodingSubset{TS1}{futj}    {4}
```

Y&Y's Lucida Bright

```
\DeclareEncodingSubset{TS1}{hlh}     {3}
\DeclareEncodingSubset{TS1}{hls}     {3}
\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`.

```
\DeclareEncodingSubset{TS1}{hlct}    {5}
\DeclareEncodingSubset{TS1}{hlx}     {5}
\DeclareEncodingSubset{TS1}{hlce}    {5}
\DeclareEncodingSubset{TS1}{hlcn}    {5}
\DeclareEncodingSubset{TS1}{hlcw}    {5}
\DeclareEncodingSubset{TS1}{hlcf}    {5}
```

Other commercial families...

```
\DeclareEncodingSubset{TS1}{pplx}    {3}
\DeclareEncodingSubset{TS1}{pplj}    {3}
\DeclareEncodingSubset{TS1}{ptmx}    {4}
\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.

```
\InputIfFileExists{textcomp.cfg}
  {\PackageInfo{textcomp}{Local configuration file used}}{}
```

```
\fi
⟨/TS1sty⟩
```

File m

ltcounts.dtx

21 Counters and Lengths

Commands for defining and using counters. This file defines:

`\newcounter` To define a new counter.

`\setcounter` To set the value of counters.

`\addtocounter` Increase the counter `#1` by the number `#2`.

`\stepcounter` Increase a counter by one.

`\refstepcounter` Increase a counter by one, also setting the value used by `\label`.

`\value` For accessing the value of the counter as a TEX number (as opposed to `\the`⟨*counter*⟩ which expands to the *printed* representation of ⟨*counter*⟩)

`\arabic` `\arabic{`⟨*counter*⟩`}`: 1, 2, 3, ...

`\roman` `\roman{`⟨*counter*⟩`}`: i, ii, iii, ...

`\Roman` `\Roman{`⟨*counter*⟩`}`: I, II, III, ...

`\alph` `\alph{`⟨*counter*⟩`}`: a, b, c, ...

`\Alph` `\Alph{`⟨*counter*⟩`}`: A, B, C, ...

`\fnsymbol` `\fnsymbol{`⟨*counter*⟩`}`: $*$, †, ‡, ...

```
⟨*2ekernel⟩
```

21.1 Environment Counter Macros

An environment foo has an associated counter defined by the following control sequences:

`\c@foo`	Contains the counter's numerical value. It is defined by `\newcount\foocounter`.

`\thefoo`	Macro that expands to the printed value of `\foocounter`. For example, if sections are numbered within chapters, and section headings look like
	Section II-3. The Nature of Counters
	then `\thesection` might be defined by:
	`\def\thesection`
	`{\@Roman{\c@chapter}-\@arabic{\c@section}}`

`\p@foo`	Macro that expands to a printed 'reference prefix' of counter foo. Any `\ref` to a value created by counter foo will produce the expansion of `\p@foo\thefoo` when the `\label` command is executed. See file `ltxref.dtx` for an extension of this mechanism.
`\cl@foo`	List of counters to be reset when foo stepped. Has format `\@elt{countera}\@elt{counterb}\@elt{counterc}`.

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*⟩.

```
\def\setcounter#1#2{%
  \@ifundefined{c@#1}%
    {\@nocounterr{#1}}%
    {\global\csname c@#1\endcsname#2\relax}}
```

`\addtocounter`

`\addtocounter{`⟨*foo*⟩`}{`⟨*val*⟩`}` Globally increments `\foocounter` by ⟨*val*⟩.

```
\def\addtocounter#1#2{%
  \@ifundefined{c@#1}%
    {\@nocounterr{#1}}%
    {\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.

```
\def\newcounter#1{%
  \expandafter\@ifdefinable \csname c@#1\endcsname
    {\@definecounter{#1}}%
  \@ifnextchar[{\@newctr{#1}}{}}
```

`\value`

`\value{`⟨*ctr*⟩`}` produces the value of counter ⟨*ctr*⟩, for use with a `\setcounter` or `\addtocounter` command.

```
\def\value#1{\csname c@#1\endcsname}
```

`\@newctr`

```
\def\@newctr#1[#2]{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}
```

`\stepcounter`

`\stepcounter`foo Globally increments counter `\c@FOO` and resets all subsidiary counters.

```
\def\stepcounter#1{%
  \addtocounter{#1}\@ne
  \begingroup
    \let\@elt\@stpelt
    \csname cl@#1\endcsname
  \endgroup}
```

`\@stpelt`

```
\def\@stpelt#1{\global\csname c@#1\endcsname \z@}
```

`\cl@@ckpt`

```
\def\cl@@ckpt{\@elt{page}}
```

`\@definecounter`

```
\def\@definecounter#1{\expandafter\newcount\csname c@#1\endcsname
     \setcounter{#1}\z@
     \global\expandafter\let\csname cl@#1\endcsname\@empty
     \@addtoreset{#1}{@ckpt}%
     \global\expandafter\let\csname p@#1\endcsname\@empty
     \expandafter
     \gdef\csname the#1\expandafter\endcsname\expandafter
          {\expandafter\@arabic\csname c@#1\endcsname}}
```

`\@addtoreset`

```
\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.

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

`\roman` Representation of ⟨*counter*⟩ as lower-case Roman numerals.

```
\def\roman#1{\expandafter\@roman\csname c@#1\endcsname}
```

`\Roman` Representation of ⟨*counter*⟩ as upper-case Roman numerals.

```
\def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}
```

`\alph` Representation of ⟨*counter*⟩ as a lower-case letter: 1 = a, 2 = b, etc.

```
\def\alph#1{\expandafter\@alph\csname c@#1\endcsname}
```

`\Alph` Representation of ⟨*counter*⟩ as an upper-case letter: 1 = A, 2 = B, etc.

```
\def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}
```

`\fnsymbol` Representation of ⟨*COUNTER*⟩ as a footnote symbol: 1 = $*$, 2 = †, etc.

```
\def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}
```

`\@arabic` `\@arabic\FOOcounter` Representation of `\FOOcounter` as arabic numerals.

```
\def\@arabic#1{\number #1}  %% changed 29 Apr 86
```

`\@roman` `\@roman\FOOcounter` Representation of `\FOOcounter` as lower-case Roman numerals.

```
\def\@roman#1{\romannumeral #1}
```

`\@Roman` `\@Roman\FOOcounter` Representation of `\FOOcounter` as upper-case Roman numerals.

```
\def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}
```

`\@slowromancap` Fully expandable macro to change a roman number to uppercase.

```
\def\@slowromancap#1{\ifx @#1% then terminate
     \else
       \if i#1I\else\if v#1V\else\if x#1X\else\if l#1L\else\if
       c#1C\else\if d#1D\else \if m#1M\else#1\fi\fi\fi\fi\fi\fi\fi
       \expandafter\@slowromancap
     \fi
}
```

`\@alph` `\@alph\FOOcounter` Representation of `\FOOcounter` as a lower-case letter: 1 = a, 2 = b, etc.

```
\def\@alph#1{%
  \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
   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
    y\or z\else\@ctrerr\fi}
```

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

```
\def\@Alph#1{%
  \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
   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
    Y\or Z\else\@ctrerr\fi}
```

`\@fnsymbol` Typesetting old fashioned footnote symbols. This can be done both in text or math mode now.

```
\def\@fnsymbol#1{\ensuremath{\ifcase#1\or *\or \dagger\or \ddagger\or
   \mathsection\or \mathparagraph\or \|\or **\or \dagger\dagger
   \or \ddagger\ddagger \else\@ctrerr\fi}}
```

⟨/2ekernel⟩

File n

ltlength.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`.

```
⟨*2ekernel⟩
\message{lengths,}
```

`\newlength`

```
\def\newlength#1{\@ifdefinable#1{\newskip#1}}
```

`\setlength`

```
\def\setlength#1#2{#1#2\relax}
```

`\addtolength` `\relax` added 24 Mar 86

```
\def\addtolength#1#2{\advance#1 #2\relax}
```

`\settoheight` `\settodepth` `\settowidth` `\@settodim` The obvious analogs of `\settowidth`.

```
\def\@settodim#1#2#3{\setbox\@tempboxa\hbox{{#3}}#2#1\@tempboxa
```

Clear the memory afterwards (which might be a lot).

```
        \setbox\@tempboxa\box\voidb@x}
\def\settoheight{\@settodim\ht}
\def\settodepth {\@settodim\dp}
\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 calulating a dimension.

```
\def\@settopoint#1{\divide#1\p@\multiply#1\p@}
⟨/2ekernel⟩
```

File o

ltfssbas.dtx

This file contains the main implementation of the 'low level' font selection commands. See other parts of the LaTeX distribution, or *The LaTeX Companion* for higher level documentation of the LaTeX '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.

23 Autoloading parts of NFSS

This code is set up in a way that some parts of it can be kept separate and will only be loaded if needed.

If we are producing an autoload version of LaTeX 2$_\varepsilon$ then all those parts with def1 or def2 docstrip guards will be placed into the autoloadable files `autofss1.sty` and `autofss2.sty`.

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.

```
⟨2ekernel⟩\expandafter\let\csname ver@autofss1.sty\endcsname\fmtversion
```

The autoload file `autofss2` is a specialty because it contains code which will be completely local, ie loaded every time again.

24 Preliminary macros

We define a number of macros that will be used later.

`\@nomath` `\@nomath` is used by most macros that will have no effect in math mode. It issues a warning message.

```
⟨*2ekernel | autoload⟩
\def\@nomath#1{\relax\ifmmode
  \@font@warning{Command \noexpand#1invalid in math mode}\fi}
⟨/2ekernel | autoload⟩
```

`\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 TeX from scanning too far in certain situations.

```
⟨*2ekernel | def1⟩
\gdef\no@alphabet@error#1{\relax \ifmmode
    \@latex@error{Math\space alphabet\space identifier\space
          \noexpand#1is\space undefined\space in\space math\space
           version\space `\math@version'}%
        {Your\space requested\space math\space alphabet\space
         is\space undefined\space in\space the\space current\space
          math\space version.^^JCheck\space the\space spelling\space
          or\space use\space the\space \noexpand\SetMathAlphabet\space
          command.}
     \fi}
⟨/2ekernel | def1⟩
⟨*autoload⟩
\gdef\no@alphabet@error{\relax \ifmmode
```

```
    \expandafter\try@sizes\expandafter\no@alphabet@error \fi}
⟨/autoload⟩
```

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

```
⟨*2ekernel | autoload⟩
\def\new@mathgroup{\alloc@8\mathgroup\chardef\sixt@@n}
\let\mathgroup\fam
\let\newfam\new@mathgroup
\@onlypreamble\new@mathgroup
```

25 Macros for setting up the tables

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

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

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

```
   \nfss@catcodes
```

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

`\DeclareFontShape`

```
   \expandafter\endgroup
   \DeclareFontShape@}
\def\DeclareFontShape@#1#2#3#4#5#6{%
   \expandafter\ifx\csname #1+#2\endcsname\relax
     \@latex@error{Font family `#1+#2' unknown}\@eha
   \else
     \expandafter
       \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
                                    \csname #5\endcsname}%
     \def\reserved@a{#6}%
     \global
     \expandafter\let\csname#5\expandafter\endcsname
        \ifx\reserved@a\@empty
          \@empty
        \else
          \reserved@a
        \fi
   \fi
  }
```

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

```
\def\DeclareFixedFont#1#2#3#4#5#6{%
   \begingroup
      \math@fontsfalse
      \every@math@size{}%
      \fontsize{#6}\z@
      \usefont{#2}{#3}{#4}{#5}%
      \global\expandafter\let\expandafter#1\the\font
   \endgroup
  }
⟨/2ekernel | autoload⟩
```

`\do@subst@correction`

```
⟨*2ekernel | autoload⟩
\def\do@subst@correction{%
      \xdef\subst@correction{%
```

---

[2] For the same reason it seems advisable to `\let\fam` and `\newfam` equal to `\relax`, but this is commented out to retain compatibility to existing style files.

```
          \font@name
          \global\expandafter\font
            \csname \curr@fontshape/\f@size\endcsname
            \noexpand\fontname\font
           \relax}%
```

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

```
       \aftergroup\subst@correction
}
```

`\DeclareFontFamily`

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

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

```
% \@tempswafalse
% \def\reserved@b{#1}%
% \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
%      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
% \cdp@list
% \if@tempswa
 \@ifundefined{T@#1}%
    {%
     \@latex@error{Encoding scheme  ‘#1’ unknown}\@eha
    }%
    {%
```

Now we have to define the macro \⟨*#1*⟩+⟨*#2*⟩ to contain `#3`. But since most of the time `#3` will be empty we use `\let` in a tricky way rather than a simple `\def` since this will save internal memory. We store the argument `#3` in a temporary macro `\reserved@a`.

```
     \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`.

```
     \global
     \expandafter\let\csname #1+#2\expandafter\endcsname
            \ifx \reserved@a\@empty
              \@empty
            \else \reserved@a
            \fi
    }%
}
```

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

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

`\cdp@elt`

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

`\DeclareFontEncoding`

```
\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.

```
   \begingroup
   \nfss@catcodes
   \expandafter\endgroup
   \DeclareFontEncoding@}
\@onlypreamble\DeclareFontEncoding
```

```
\def\DeclareFontEncoding@#1#2#3{%
  \expandafter
  \ifx\csname T@#1\endcsname\relax
     \def\cdp@elt{\noexpand\cdp@elt}%
     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
                    {\default@family}{\default@series}%
                    {\default@shape}}%
```

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

```
     \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
  \else
     \@font@info{Redeclaring font encoding #1}%
  \fi
```

```
  \global\@namedef{T@#1}{#2}%
  \global\@namedef{M@#1}{\default@M#3}%
```

Keep a record of the last encoding being declared:

```
  \xdef\LastDeclaredEncoding{#1}%
  }
\@onlypreamble\DeclareFontEncoding@
```

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

```
\def\LastDeclaredEncoding{}
```

`\DeclareFontSubstitution`

```
\def\DeclareFontSubstitution#1#2#3#4{%
  \expandafter
  \ifx\csname T@#1\endcsname\relax
    \@latex@error{Encoding scheme  `#1' unknown}\@eha
  \else
    \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`.

```
       \edef\reserved@a{#1}%
       \toks@{}%
       \def\cdp@elt##1##2##3##4{%
          \def\reserved@b{##1}%
          \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.

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

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

```
            \addto@hook\toks@{\cdp@elt{##1}{##2}{##3}{##4}}%
          \fi}%
        \cdp@list
        \xdef\cdp@list{\the\toks@}%
    \endgroup
    \global
    \@namedef{D@#1}{%
           \def\default@family{#2}%
           \def\default@series{#3}%
           \def\default@shape{#4}%
           }%
  \fi
 }
\@onlypreamble\DeclareFontSubstitution
```

`\DeclareFontEncodingDefaults`

```
\def\DeclareFontEncodingDefaults#1#2{%
  \ifx\relax#1\else
    \ifx\default@T\@empty\else
      \@font@info{Overwriting encoding scheme text defaults}%
    \fi
    \gdef\default@T{#1}%
  \fi
  \ifx\relax#2\else
    \ifx\default@M\@empty\else
      \@font@info{Overwriting encoding scheme math defaults}%
    \fi
    \gdef\default@M{#2}%
  \fi
}
\@onlypreamble\DeclareFontEncodingDefaults
```

`\default@T`

`\default@M`

```
\let\default@T\@empty
\let\default@M\@empty
```

`\DeclarePreloadSizes`

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

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

```
   \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).

```
    \def\reserved@f##1,{%
```

The end of the list will be detected when there are no more elements, i.e. when `\reserved@f`'s argument is empty. The trick used here is explained in Appendix D of the TEXbook: if the argument is empty the `\if` will select the first clause and `\let` `\reserved@f` equal to `\relax`. (We use the `>` character here since it cannot appear in font file names.)

```
        \if>##1>%
          \let\reserved@f\relax
        \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.

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

---

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

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

```
        \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.

```
     \reserved@f#5,,%
   \endgroup
   }%
}
\@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.

```
\newif\ifmath@fonts \math@fontstrue
```

`\DeclareMathSizes`
`\DeclareMathSizes*`

`\DeclareMathSizes` takes the text size, math text size, math script size, and math scriptscript size as arguments and defines the right `\S@`... macro.

```
\def\DeclareMathSizes{%
  \@ifstar{\@DeclareMathSizes\math@fontsfalse}%
          {\@DeclareMathSizes{}}}
\@onlypreamble\DeclareMathSizes
```

`\@DeclareMathSizes`

```
\def\@DeclareMathSizes#1#2#3#4#5{%
   \@defaultunits\dimen@#2pt\relax\@nnil
    \if$#3$%
      \expandafter \let
        \csname S@\strip@pt\dimen@\endcsname
        \math@fontsfalse
    \else
      \expandafter \gdef
      \csname S@\strip@pt\dimen@\endcsname
            {\gdef\tf@size{#3}\gdef\sf@size{#4}%
                             \gdef\ssf@size{#5}%
             #1%
                             }%
    \fi}
\@onlypreamble\@DeclareMathSizes
```

26 Selecting a new font

26.1 Macros for the user

`\fontencoding`
`\f@encoding`

As we said in the introduction a font is described by four parameters. We first define macros to specify the wanted *family*, *series*, or *shape*. These are simply recorded in internal macros `\f@family`, `\f@series`, and `\f@shape`, resp. We use `\edef`'s so that the arguments can also be macros.

```
\DeclareRobustCommand\fontencoding[1]{%
    \expandafter\ifx\csname T@#1\endcsname\relax
      \@latex@error{Encoding scheme `#1' unknown}\@eha
    \else
      \edef\f@encoding{#1}%
      \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` inbetween we can save processing by making sure that `\enc@update` is `\relax`.

```
        \let\enc@update\relax
      \else
```

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

```
        \let\enc@update\@@enc@update
      \fi
    \fi
}
```

`\@@enc@update`

```
\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`.

```
          \expandafter
          \let
            \csname\cf@encoding -cmd\endcsname
            \@changed@cmd
```

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

```
          \expandafter
          \let
            \csname\f@encoding-cmd\endcsname
            \@current@cmd
```

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

```
        \default@T
        \csname T@\f@encoding\endcsname
```

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

```
        \csname D@\f@encoding\endcsname
        \let\enc@update\relax
        \let\cf@encoding\f@encoding
}
```

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

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

`\fontfamily`
`\f@family`
`\fontseries`
`\f@series`
`\fontshape`
`\f@shape`

```
\DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
```

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` comand first.

```
\def\usefont#1#2#3#4{\fontencoding{#1}\fontfamily{#2}%
            \fontseries{#3}\fontshape{#4}\selectfont
            \ignorespaces}
```

`\linespread` The comand `\linespread` changes the current `\baselinestretch` by calling `\set@fontsize`. The values for `\f@size` and `\f@baselineskip` will be left unchanged.

```
\DeclareRobustCommand\linespread[1]
   {\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 efectively overwritten by a size change.

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

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

```
\let\f@family\@empty
\let\f@series\@empty
\let\f@shape\@empty
\let\f@size\@empty
\let\f@baselineskip\@empty
\let\f@linespread\@empty
```

`\cf@encoding`

```
\let\f@encoding\@empty
\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.

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

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

`\rem@pt`

```
\begingroup
  \catcode`P=12
  \catcode`T=12
  \lowercase{
    \def\x{\def\rem@pt##1.##2PT{##1\ifnum##2>\z@.##2\fi}}}
  \expandafter\endgroup\x
\def\strip@pt{\expandafter\rem@pt\the}
```

`\mathversion` `\math@version` `\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.

```
\DeclareRobustCommand\mathversion[1]
        {\@nomath\mathversion
         \expandafter\ifx\csname mv@#1\endcsname\relax
         \@latex@error{Math version `#1' is not defined}\@eha\else
         \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@currsize` to an invalid value since this will trigger the setup when the formula starts.

```
         \gdef\glb@currsize{}%
```

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 todo the setting only if the version really has changed after all. This might be interesting in case of `amstext` and `boldsymbol`.

```
           \aftergroup\glb@settings
           \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$_\varepsilon$ 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`
`\frozen@everydisplay`

New internal names for `\everymath` and `\everydisplay`.

```
\let\frozen@everymath\everymath
\let\frozen@everydisplay\everydisplay
```

`\everymath`
`\everydisplay`

Now we provide now user hooks that will be called in the frozen internals.

```
\newtoks\everymath
\newtoks\everydisplay
```

`\frozen@everymath`

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

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

`\frozen@everydisplay`

Ditto for the display hook.

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

`\curr@math@size`

This holds locally the current math size.

```
\let\curr@math@size\@empty
```

26.2 Macros for loading fonts

`\pickup@font`

The macro `\pickup@font` which is used in `\selectfont` is very simple: if the font name is undefined (i.e. not known yet) it calls `\define@newfont` to load it.

```
\def\pickup@font{%
    \expandafter \ifx \font@name \relax
       \define@newfont
    \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`

```
{\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.

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

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

```
\def\curr@fontshape{\f@encoding/\f@family/\f@series/\f@shape}
⟨/2ekernel | autoload⟩
```

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

```
⟨*2ekernel | def2 | autoload⟩
\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.

```
  \begingroup
    \let\typeout\@font@info
    \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`.

```
    \expandafter\expandafter\expandafter
       \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.

```
%    \expandafter\ifx
%        \csname\curr@fontshape\endcsname \relax
      \try@load@fontshape % try always
%    \fi
    \expandafter\ifx
       \csname\curr@fontshape\endcsname \relax
      \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).

```
%      \csname\curr@fontshape\endcsname
      \extract@font\fi
```

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

```
  \endgroup}
⟨/2ekernel | def2 | autoload⟩
```

As `autofss2.sty` only makes local definitions it is re-loaded for each font, to save some string memory in the kernel, and to speed up the loading of some packages which may load fonts The code is actually pre-loaded into the kernel and removed at `\begin{document}`. The `\ifx` test below ensures that if `\usepackage{autofss2}` apears in the preamble, then the code is not removed at this time. Can not use `\AtBeginDocument` here as it is not defined yet! Listing all the commands like this is not ideal as any changes to the `autofss2.sty` need to be reflected here, but this seems the most memory efficient mechanism as it avoids the use of an extra csname to store the list.

This is currently disabled, so the ‘autofss2’ code remains in the kernel, and `autofss2.sty` is not generated in the current public release.

```
⟨*autoloadxxx⟩
\expandafter\def\expandafter\@begindocumenthook\expandafter{%
  \expandafter\ifx\csname ver@autofss2.sty\endcsname\relax
  \gdef\define@newfont{%
    \begingroup
      \makeatletter\nfss@catcodes
      \catcode`\#6\relax
      \@@input autofss2.sty\relax\define@newfont
    \endgroup}%
  \begingroup
    \def\do##1{\global\let##1\@undefined}%
    \do\extract@sizefn
    \do\try@simple@size
    \do\set@simple@size@args
    \do\extract@rangefontinfo
    \do\is@range
    \do\check@range
    \do\check@single
    \do\set@size@funct@args
    \do\set@size@funct@args@
    \do\try@size@range
    \do\empty@sfcnt
    \do\gen@sfcnt
    \do\genb@sfcnt
    \do\sub@sfcnt
    \do\subf@sfcnt
    \do\fixed@sfcnt
  \endgroup
  \fi}
⟨/autoloadxxx⟩
```

```
⟨*2ekernel | autoload⟩
\def\try@load@fontshape{%
   \expandafter
   \ifx\csname \f@encoding+\f@family\endcsname\relax
     \@font@info{Try loading font information for
                   \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.

```
    \global\expandafter\let
       \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).

```
     \nfss@catcodes
     \let\nfss@catcodes\relax
```

For increased portability make the external filename monocase, but look for the (old style) mixed case filename if the first attempt fails.

On any monocase 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.

```
     \edef\reserved@a{%
       \lowercase{%
         \noexpand\InputIfFileExists{\f@encoding\f@family.fd}}}%
     \reserved@a\relax
          {\@input@{\f@encoding\f@family.fd}}%
```

```
   \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 `\catcode`s 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.

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

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

```
     \makeatletter
     \catcode`\ 9%
     \catcode`\^^I9%
     \catcode`\^^M9%
```

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

```
     \catcode`\\\z@
     \catcode`\{\@ne
     \catcode`\}\tw@
     \catcode`\#6%
     \catcode`\^7%
     \catcode`\%14%
```

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

```
   \@makeother\<%
   \@makeother\>%
   \@makeother\*%
   \@makeother\.%
   \@makeother\-%
   \@makeother\/%
   \@makeother\[%
   \@makeother\]%
   \@makeother\`%
   \@makeother\'%
   \@makeother\"%
}
```

\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!

```
\def\DeclareErrorFont#1#2#3#4#5{%
      \xdef\error@fontshape{%
          \noexpand\expandafter\noexpand\split@name\noexpand\string
          \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
          \noexpand\@nil}%
```

Initialize all those internal variables which may or may not have values in the first seconds of NFSS' bootstraping 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.

```
%      \gdef\f@encoding{#1}%
      \gdef\default@family{#2}%
      \gdef\default@series{#3}%
      \gdef\default@shape{#4}%
      \global\let\f@family\default@family
      \global\let\f@series\default@series
      \global\let\f@shape\default@shape
      \gdef\f@size{#5}%
      \gdef\f@baselineskip{#5pt}%
}
\@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!

```
\def\wrong@fontshape{%
    \csname D@\f@encoding\endcsname      % install defaults if in math
```

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

```
    \edef\reserved@a{\csname\curr@fontshape\endcsname}%
  \ifx\last@fontshape\reserved@a
    \errmessage{Corrupted NFSS tables}%
    \error@fontshape
  \else
```

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

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

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

```
    \expandafter\ifx\csname\curr@fontshape\endcsname\relax
       \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!

```
        \expandafter
          \ifx\csname\curr@fontshape\endcsname\relax
           \let\f@family\default@family
        \fi \fi
  \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.

```
     \@font@warning{Font shape `\expandafter\string\reserved@a'
                     \expandafter\@gobble\string\@undefined\MessageBreak
                   using `\curr@fontshape' instead\@wrong@font@char}%
    \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`.

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

```
    \global\expandafter\expandafter\expandafter\let
       \expandafter\reserved@a
          \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.

```
    \xdef\font@name{%
      \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.

```
    \pickup@font}
```

`\@wrong@font@char` Normally empty but redefined in `\UseTextSymbol` so that the Font shape undefined message can refer to the symbol causing the problem.

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

`\@@defaultsubs` See above.

`\@defaultsubs`

```
\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).

```
\def\strip@prefix#1>{}
```

27 Assigning math fonts to *versions*

`\install@mathalphabet` This is just another name for `\gdef` but we can redefine it if necessary later on.

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

`\math@fonts`

```
\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.

```
%\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.

```
%  \begingroup
```

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

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

We globally select the math fonts…

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

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

```
%  \endgroup
```

As long as no *size* or *version* change occurs the ⟨*math alphabet identifier*⟩ should simply switch to the installed *math group* instead of calling `\select@group` unnecessarily. So we globally redefine the first argument (the new ⟨*math alphabet identifier*⟩) to expand into a `\mathgroup` switch and then select this *alphabet.* Note that this redefinition will be overwritten by the next call to a *version* macro.
The original code for the end of `\select@group` was

```
\gdef#1{#3\mathgroup #2}#1\fi}
```

i.e. first redefining the ⟨*math alphabet identifier*⟩ and then calling the new definition to switch to the wanted ⟨*math group*⟩. Now we define the ⟨*math alphabet identifier*⟩ as a call to the `\use@mathgroup` command.

```
%   \xdef#1{\noexpand\use@mathgroup\noexpand#2%
%           {\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 identifer `#1` and the math version macro as arguments.

```
%     \expandafter\extract@alph@from@version
%        \csname mv@\math@version\expandafter\endcsname
%       \expandafter{\number\csname c@mv@\math@version\endcsname}%
%       #1%
%     \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 od `\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.

```
%\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.

```
\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`.

```
      \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`.

```
          \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`.

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

Then we execute our auxiliary macro.

```
      \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.

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

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

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

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

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

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

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

`\math@bgroup`
`\math@egroup`

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.

```
\let\math@bgroup\bgroup
\def\math@egroup#1{#1\egroup}
⟨/2ekernel | autoload⟩
```

`\calculate@math@sizes`

Here is the default definition for `\calculate@math@sizes` a more elaborate interface is under testing in mthscale.sty.

```
⟨*2ekernel | def1⟩
\gdef\calculate@math@sizes{%
  \@font@info{Calculating\space math\space sizes\space for\space
              size\space <\f@size>}%
  \dimen@\f@size \p@
  \@tempdimb \defaultscriptratio \dimen@
  \dimen@ \defaultscriptscriptratio \dimen@
  \expandafter\xdef\csname S@\f@size\endcsname{%
    \gdef\noexpand\tf@size{\f@size}%
    \gdef\noexpand\sf@size{\strip@pt\@tempdimb}%
    \gdef\noexpand\ssf@size{\strip@pt\dimen@}%
    \noexpand\math@fontstrue}}
⟨/2ekernel | def1⟩
⟨*autoload⟩
\def\calculate@math@sizes{\try@sizes\calculate@math@sizes}
⟨/autoload⟩
```

`\defaultscriptratio`
`\defaultscriptscriptratio`

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

```
⟨*2ekernel | autoload⟩
\def\defaultscriptratio{.7}
\def\defaultscriptscriptratio{.5}
```

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

```
\ifx\noaccents@\@undefined
  \let\noaccents@\@empty
\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 LaTeX.

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoerr⟩
\gdef\showhyphens#1{%
  \setbox0\vbox{%
    \color@begingroup
    \everypar{}%
    \parfillskip\z@skip\hsize\maxdimen
    \normalfont
    \pretolerance\m@ne\tolerance\m@ne\hbadness\z@\showboxdepth\z@\ #1%
    \color@endgroup}}
⟨/2ekernel | autoerr⟩
⟨autoload⟩\def\showhyphens{\@autoerr\showhyphens}
⟨*2ekernel | autoload⟩
```

`\addto@hook` We need a macro to add tokens to a hook.

```
\long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}
```

`\@vpt`

```
 \def\@vpt{5}
```

`\@vipt`

```
 \def\@vipt{6}
```

`\@viipt`

```
 \def\@viipt{7}
```

`\@viiipt`

```
 \def\@viiipt{8}
```

`\@ixpt`

```
 \def\@ixpt{9}
```

`\@xpt`

```
 \def\@xpt{10}
```

`\@xipt`

```
 \def\@xipt{10.95}
```

`\@xiipt`

```
 \def\@xiipt{12}
```

`\@xivpt`

```
 \def\@xivpt{14.4}
```

`\@xviipt`

```
 \def\@xviipt{17.28}
```

`\@xxpt`

```
 \def\@xxpt{20.74}
```

`\@xxvpt`

```
\def\@xxvpt{24.88}
⟨/2ekernel | autoload⟩
```

File p

ltfsstrc.dtx

28 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 LaTeX will stop.

29 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.

When this file is processed directly by LaTeX this will produce the documentation as well.

```
⟨*driver⟩
\documentclass{ltxdoc}


%\OnlyDescription  % comment out for implementation details

\begin{document}
   \DocInput{ltfsstrc.dtx}
\end{document}
⟨/driver⟩
```

30 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.

```
⟨*package⟩
%\NeedsTeXFormat{LaTeX2e}
%\ProvidesPackage{tracefnt}[??/??/?? v?.??
%                           Standard LaTeX package (font tracing)]
⟨/package⟩
```

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

```
⟨+debug⟩ \input trace.sty
```

31 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.

```
⟨*2ekernel | autoload⟩
\def\tracingfonts{%
  \@font@warning{Command \noexpand\tracingfonts
           not provided.\MessageBreak
           Use the ‘tracefnt’ package.\MessageBreak Command found:}%
       \count@}
⟨/2ekernel | autoload⟩
```

The `\count@` in the line above will remove the number after `\tracingfonts`. Note that this definition will be overwritten be the next line if one of these modules are included.

```
⟨*package, trace, debug⟩
\newcount\tracingfonts
\tracingfonts=0
⟨/package, trace, debug⟩
```

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

```
⟨*package⟩
\DeclareOption{errorshow}{%
   \def\@font@info#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
    \def\@font@warning#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                        {LaTeX Font Warning: #1}}%
     }

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

\DeclareOption{infoshow}{%
   \def\@font@info#1{%
         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
```

---

[4] 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.

```
    \def\@font@warning#1{%
         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                         {LaTeX Font Warning: #1}}%
     }
\DeclareOption{loading}{%
    \tracingfonts\tw@
   }
\DeclareOption{debugshow}{%
    \ExecuteOptions{infoshow}%
    \tracingfonts\thr@@
   }

\DeclareOption{pausing}{%
    \def\@font@warning#1{%
      \GenericError
             {(Font)\@spaces\@spaces\@spaces\space\space}%
             {LaTeX Font Warning: #1}%
             {See the LaTeX Companion for details.}%
             {I'll stop for every LaTeX Font Warning because
              you requested\MessageBreak the `pausing' option
              to the tracefnt package.}}%
   }
```

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

```
\ExecuteOptions{infoshow}
\ProcessOptions
⟨/package⟩
```

We also need a default definition inside the kernel:

```
⟨*2ekernel | autoload⟩
\def\@font@info#1{%
         \GenericInfo{(Font)\@spaces\@spaces\@spaces\space\space}%
                     {LaTeX Font Info: \space\space\space#1}}%
\def\@font@warning#1{%
         \GenericWarning{(Font)\@spaces\@spaces\@spaces\space\space}%
                         {LaTeX Font Warning: #1}}%
⟨/2ekernel | autoload⟩
```

32 Macros common to `fam.tex` and `tracefnt.sty`

In the first versions of `tracefnt.dtx` some macros of `fam.dtx`[5] were redefined to included the extra tracing information. Now these macros are all defined in this file (i.e. removed from `fam.dtx`) and different production versions can be obtained simply by specifying a different set of modules to include when generating `ltfss.dtx`.

32.1 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.

```
⟨*2ekernel | package | autoload⟩
\def\extract@font{%
   \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`).

```
   \global\expandafter\font\font@name\external@font\relax
```

---

[5] This file is currently not distributed in documented form. Its code is part of `ltfss.dtx`.

When tracing we typeout the internal and external font name.

```
⟨*trace⟩
    \ifnum \tracingfonts >\@ne
    \@font@info{External font `\external@font'
                loaded as\MessageBreak \font@name}\fi
⟨/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.

```
    \font@name \relax
```

The next two lines execute the “loading actions” for the family and then for the individual font shape.

```
    \csname \f@encoding+\f@family\endcsname
    \csname\curr@fontshape\endcsname
    \relax
       }
⟨/2ekernel | package | autoload⟩
```

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`.

```
⟨*2ekernel | autoload⟩
\def\get@external@font{%
```

We don't know the external font name at the beginning.

```
   \let\external@font\@empty
   \edef\font@info{\expandafter\expandafter\expandafter\string
       \csname \curr@fontshape \endcsname}%
   \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.

```
   \ifx\external@font\@empty
      \try@size@substitution
      \ifx\external@font\@empty
         \@latex@error{Font \expandafter \string\font@name\space
                     not found}\@eha
         \error@fontshape
         \get@external@font
   \fi\fi
}
⟨/2ekernel | autoload⟩
```

`\selectfont` The macro `\selectfont` is called whenever a font change must take place.

```
⟨*2ekernel | package | autoload⟩
\DeclareRobustCommand\selectfont
        {%
```

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.

```
⟨+debug⟩  \pushtracing
⟨+debug⟩  \ifnum\tracingfonts<4 \tracingoff
⟨+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`.

```
    \ifx\f@linespread\baselinestretch \else
      \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 LaTeX's `\twfbf`, etc. We define `\font@name` globally, as always. The reason for this is explained later on.

```
    \xdef\font@name{%
      \csname\curr@fontshape/\f@size\endcsname}%
```

We call the macro `\pickup@font` which will load the font if necessary.

```
    \pickup@font
```

Then we select the font.

```
    \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`.

```
⟨*trace⟩
    \ifnum \tracingfonts>\tw@
      \@font@info{Switching to \font@name}\fi
⟨/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.

```
    \size@update
```

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

```
    \enc@update
```

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

```
⟨+debug⟩ \poptracing
    }
```

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

```
\def\set@fontsize#1#2#3{%
    \@defaultunits\@tempdimb#2pt\relax\@nnil
    \edef\f@size{\strip@pt\@tempdimb}%
    \@defaultunits\@tempskipa#3pt\relax\@nnil
    \edef\f@baselineskip{\the\@tempskipa}%
    \edef\f@linespread{#1}%
```

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

```
    \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:

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

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

```
        \baselineskip\f@baselineskip\relax
        \baselineskip\f@linespread\baselineskip
        \normalbaselineskip\baselineskip
```

then to set up a new `\strutbox`

```
        \setbox\strutbox\hbox{%
          \vrule\@height.7\baselineskip
                \@depth.3\baselineskip
                \@width\z@}%
```

We end with a bit of tracing information.

```
⟨*trace⟩
  \ifnum \tracingfonts>\tw@
     \ifx\f@linespread\@empty
       \let\reserved@a\@empty
     \else
       \def\reserved@a{\f@linespread x}%
     \fi
     \@font@info{Changing size to \f@size/\reserved@a
                \f@baselineskip}%
     \aftergroup\type@restoreinfo \fi
⟨/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.

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

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.

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

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

```
⟨*trace⟩
  \def\type@restoreinfo{%
     \ifx\f@linespread\@empty
       \let\reserved@a\@empty
     \else
       \def\reserved@a{\f@linespread x}%
     \fi
     \@font@info{Restoring size to
                \f@size/\reserved@a\f@baselineskip}}
⟨/trace⟩
```

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

```
\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.

```
    \expandafter\ifx\csname S@\f@size\endcsname\relax
      \calculate@math@sizes
    \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.

```
    \csname S@\f@size\endcsname
    \ifmath@fonts
⟨*trace⟩
      \ifnum \tracingfonts>\tw@
```

```
        \@font@info{Setting up math fonts for
                  \f@size/\f@baselineskip}\fi
⟨/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.

```
      \begingroup
        \escapechar\m@ne
        \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@currsize` is set equal to `\f@size`. This signals that the fonts for math in this size are set up.

```
        \globaldefs\@ne
        \math@fonts
        \let \glb@currsize \f@size
      \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.

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

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

```
⟨*trace⟩
      \else
        \ifnum \tracingfonts>\tw@
          \@font@info{No math setup for
                    \f@size/\f@baselineskip}\fi
⟨/trace⟩
    \fi
}
⟨/2ekernel | package | autoload⟩
```

`\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).

```
⟨*2ekernel | autoload⟩
\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.

```
\newtoks\every@math@size
\every@math@size={}
⟨/2ekernel | autoload⟩
```

32.2 Math fonts setup

32.2.1 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 typesetable. 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 ($2 \times \langle$non-math levels$\rangle$ per inner formula).

32.2.2 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.

```
⟨*2ekernel | package | autoload⟩
\def\check@mathfonts{%
  \ifx \glb@currsize \f@size
⟨*trace⟩
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: no change \f@size\space
         curr/global (\curr@math@size/\glb@currsize)}\fi
⟨/trace⟩
  \else
⟨*trace⟩
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: setting up \f@size\space
         curr/global (\curr@math@size/\glb@currsize)}\fi
⟨/trace⟩
     \glb@settings
     \init@restore@glb@settings
  \fi
  \let\curr@math@size\f@size
  \def\init@restore@glb@settings{\aftergroup\restglb@settings}%
}
```

`\init@restore@glb@settings`

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

```
⟨-trace⟩\let\init@restore@glb@settings\relax
⟨*trace⟩
\def\init@restore@glb@settings{%
       \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: no resetting (not in
                    nested math)}\fi
}
⟨/trace⟩
```

`\restglb@settings`

This macro will be executed the first time after the current formula.

```
\def\restglb@settings{%
⟨*trace⟩
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: restoring}\fi
⟨/trace⟩
       \begingroup
        \let\f@size\curr@math@size
        \ifx\glb@currsize \f@size
⟨*trace⟩
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: ... already okay (\f@size)}\fi
⟨/trace⟩
        \else
⟨*trace⟩
    \ifnum \tracingfonts>\thr@@
        \@font@info{*** MATH: ... to \f@size}\fi
⟨/trace⟩
          \glb@settings
        \fi
      \endgroup
}
```

32.2.3 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.

```
\def\use@mathgroup#1#2{\relax\ifmmode
```

```
⟨*trace⟩
  \ifnum \tracingfonts>\tw@
    \count@#2\relax
    \@font@info{Using \noexpand\mathgroup
             (\the\count@) #2}\fi
⟨/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.

```
    \math@bgroup
        \expandafter\ifx\csname M@\f@encoding\endcsname#1\else
        #1\fi
        \mathgroup#2\relax
```

Before we reinsert the swallowed token (arg. three) into the input stream, in the case that the ⟨*math alphabet identifier*⟩ isn't called in math mode, we remove the `\fi` with the `\expandafter` trick. This is necessary if the token is actually an macro with arguments. In such a case the `\fi` will be misinterpreted as the first argument which would be disastrous.

```
    \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 AMS-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`.

```
⟨*trace⟩
  \ifx\math@bgroup\bgroup
    \def\math@egroup#1{#1\egroup
      \ifnum \tracingfonts>\tw@
     \@font@info{Restoring \noexpand\mathgroup
        (\ifnum\mathgroup=\m@ne default\else \the\mathgroup \fi)%
        }\fi}
  \fi
⟨/trace⟩
```

`\getanddefine@fonts` `\getanddefine@fonts` has two arguments: the ⟨*math group number*⟩ and the *family/series/shape* name as a control sequence.

```
\def\getanddefine@fonts#1#2{%
```

First we turn of tracing when `\tracingfonts` is less than 4.

```
⟨+debug⟩   \pushtracing
⟨+debug⟩   \ifnum\tracingfonts<4 \tracingoff
⟨+debug⟩   \else \tracingon\getanddefine@fonts \fi

⟨*trace⟩
  \ifnum \tracingfonts>\tw@
  \count@#1\relax
    \@font@info{\noexpand\mathgroup (\the\count@) #1 :=\MessageBreak
               \string#2 \tf@size/\sf@size/\ssf@size}\fi
⟨/trace⟩
```

We append the current `\tf@size` to `#2` to obtain the font name.[6] Again, `font@name` is defined globally, for the reasons explained in the description of `\wrong@fontshape`.

```
  \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`.

```
  \pickup@font \let\textfont@name\font@name
```

---

[6] 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.

Same game for `\scriptfont` and `\scriptscriptfont`:

```
\xdef\font@name{\csname \string#2/\sf@size\endcsname}%
\pickup@font \let\scriptfont@name\font@name
\xdef\font@name{\csname \string#2/\ssf@size\endcsname}%
\pickup@font
```

Then we append the new `\textfont`... assignments to the `\math@fonts`.

```
\edef\math@fonts{\math@fonts
             \textfont#1\textfont@name
             \scriptfont#1\scriptfont@name
             \scriptscriptfont#1\font@name}%
```

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

```
⟨+debug⟩  \poptracing
   }
⟨/2ekernel | package | autoload⟩
```

33 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 it argument.

```
⟨*2ekernel | autoload⟩
\def\ifnot@nil#1{\def\reserved@a{#1}%
  \ifx\reserved@a\@nnil \expandafter\@gobble
  \else \expandafter\@firstofone\fi}
```

`\remove@to@nnil`
`\remove@angles`
`\remove@star`

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`.

```
\def\remove@to@nnil#1\@nnil{}
\def\remove@angles#1>{\set@simple@size@args}
\def\remove@star#1*{#1}
⟨/2ekernel | autoload⟩
```

`\extract@sizefn` This macro takes a size specification and parses it into size function and the optional and mandatory arguments.

```
⟨*2ekernel | def2 | autoload⟩
\def\extract@sizefn#1*#2\@nil{%
  \if>#2>\set@size@funct@args#1\@nil
        \let\sizefn@info\@empty
  \else\expandafter\set@size@funct@args\remove@star#2\@nil
       \def\sizefn@info{#1}\fi
  }
```

`\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 correspondign 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`.

```
% % this could be replaced by \try@size@range making the subst slower!
\def\try@simple@size{%
```

`\reserved@a` is made an abbreviation for the head of the definition of the macro `\extract@fontinfo`.

```
     \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.

```
    \expandafter\reserved@a\expandafter<\f@size>##2<##3\@nnil{%
         \ifnot@nil{##2}%
           {\set@simple@size@args##2<##3\@nnil
            \execute@size@function\sizefn@info
           }}%
```

Now we call `\extract@fontinfo`. Note the `<\@nil tokens` at the end.

```
    \expandafter\expandafter
    \expandafter\extract@fontinfo\expandafter\font@info
    \expandafter<\f@size>\@nil<\@nnil
}
```

`\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,

```
\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.

```
         \if<#1<%
           \expandafter\remove@angles
         \else
           \extract@sizefn#1*\@nil
           \expandafter\remove@to@nnil
         \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 macors 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.

```
\def\extract@rangefontinfo#1<#2>{%
      \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.

```
\def\is@range#1-#2\@nil{%
   \if>#2\expandafter\check@single\else
      \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.

```
\def\check@range#1-#2>#3<#4\@nnil{%
  \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.

```
     \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.

```
     \upper@bound0#2\p@
     \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.

```
     \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.

```
       \lower@bound0#1\p@
       \ifdim \f@size \p@<\lower@bound
        \else
```

If both tests are passed we can try executing the size function.

```
          \set@simple@size@args#3<#4\@nnil
          \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.

```
         \ifx\external@font\@empty
         \else
           \let\reserved@f\@empty
         \fi
       \fi
     \fi
     \reserved@f}}
⟨/2ekernel | def2 | autoload⟩
```

`\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.

```
⟨*2ekernel | autoload⟩
\newdimen\lower@bound
\newdimen\upper@bound
⟨/2ekernel | autoload⟩
```

`\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).

```
⟨*2ekernel | def2 | autoload⟩
\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.

```
     \def\reserved@f{\extract@rangefontinfo<#3\@nnil}%
```

Now we check the the size against `\f@size`. If it is not equal `\f@size` it is no good and we have to recurse.

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

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

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

```
        \ifx\external@font\@empty
        \else
          \let\reserved@f\@empty
        \fi
     \fi
     \reserved@f}
```

`\set@size@funct@args`
`\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`.

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

```
\def\set@size@funct@args@[#1]#2\@nil{%
  \def\mandatory@arg{#2}%
  \def\optional@arg{#1}}
⟨/2ekernel | def2 | autoload⟩
```

`\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 `<...>`.

```
⟨*2ekernel | autoload⟩
\def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
\@onlypreamble\DeclareSizeFunction
⟨/2ekernel | autoload⟩
```

`\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`).

```
⟨*2ekernel | package | autoload⟩
\def\execute@size@function#1{%    %%% could be added to autoload as well
⟨*trace⟩
         \@ifundefined{s@fct@#1}%
           {\errmessage{Undefined font size function #1}%
            \s@fct@}%
           {\csname s@fct@#1\endcsname}%
⟨/trace⟩
⟨-trace⟩      \csname s@fct@#1\endcsname
}
⟨/2ekernel | package | autoload⟩
```

`\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.

```
⟨*2ekernel | def2 | autoload⟩
\def\try@size@range{%
     \expandafter\extract@rangefontinfo\font@info <-*>\@nil<\@nnil
}
⟨/2ekernel | def2 | autoload⟩
```

`\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.

```
⟨*2ekernel | def1⟩
\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.

```
  \@tempdimb \maxdimen
  \let \best@size \@empty
```

Now we loop over the specification

```
  \expandafter \try@simples \font@info <\number\@M>\@nil<\@nnil
}
⟨/2ekernel | def1⟩
⟨*autoload⟩
\def\try@size@substitution{\try@simples\try@size@substitution}
⟨/autoload⟩
```

`\font@submax`
`\fontsubfuzz` 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{coument}`. The macro `\fontsubfuzz` contains the amount that will not cause terminal warnings (warnings still go into the transcript file).

```
⟨*2ekernel | autoload⟩
\def\font@submax{0pt}
\def\fontsubfuzz{.4pt}
⟨/2ekernel | autoload⟩
⟨+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.

```
⟨*2ekernel | def1⟩
\gdef\try@simples#1<#2>{%
  \tryif@simple#2->\tryif@simple}
⟨/2ekernel | def1⟩
⟨*autoload⟩
\def\try@simples{\@autoload{fss1}}
⟨/autoload⟩
```

`\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.

```
⟨*2ekernel | def1⟩
\gdef\tryif@simple#1-#2\tryif@simple{%
```

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

```
  \let \reserved@f \try@simples
  \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.

```
    \dimen@ #1\p@
    \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.

```
      \ifdim \f@size\p@<\dimen@
        \@tempdimc \dimen@
        \advance\@tempdimc -\f@size\p@
      \else
        \@tempdimc \f@size\p@
        \advance\@tempdimc -\dimen@
      \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`.

```
      \ifdim \@tempdimc<\@tempdimb
        \@tempdimb \@tempdimc
        \def \best@size{#1}%
      \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`.

```
    \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.

```
% %\subst@size             %% coded inline
% %\def\subst@size{%
  \ifx \external@font\@empty
    \ifx \best@size\@empty
    \else
      \ifdim \@tempdimb>\font@submax \relax
        \xdef \font@submax {\the\@tempdimb}%
      \fi
      \let \f@user@size \f@size
      \let \f@size \best@size
      \ifdim \@tempdimb>\fontsubfuzz\relax
        \@font@warning{Font\space shape\space
            `\curr@fontshape'\space in\space size\space
             <\f@user@size>\space not\space available\MessageBreak
             size\space <\f@size>\space substituted}%
      \fi
      \try@simple@size
      \do@subst@correction
    \fi
  \fi
% %}
```

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

```
      \let \reserved@f \remove@to@nnil
    \fi
  \fi
```

If it's a range iterate also.

```
  \reserved@f}
⟨/2ekernel | def1⟩
```

33.1 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, optinal 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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{}{\empty@sfcnt\@font@warning}
\DeclareSizeFunction{s}{\empty@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\empty@sfcnt#1{%
      \@tempdimb \f@size\p@
      \ifx\optional@arg\@empty
      \else
        \@tempdimb \optional@arg\@tempdimb
        #1{Font\space shape\space `\curr@fontshape'\space
           will\space be\MessageBreak
           scaled\space to\space size\space \the\@tempdimb}%
      \fi
      \edef\external@font{\mandatory@arg\space at\the\@tempdimb}}
⟨/2ekernel | def2 | autoload⟩
```

`\s@fct@gen` `\s@fct@sgen` 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 optinal 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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{gen}{\gen@sfcnt\@font@warning}
\DeclareSizeFunction{sgen}{\gen@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\gen@sfcnt{%
      \edef\mandatory@arg{\mandatory@arg\f@size}%
      \empty@sfcnt}
⟨/2ekernel | def2 | autoload⟩
```

`\s@fct@genb` `\s@fct@sgenb` 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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{genb}{\genb@sfcnt\@font@warning}
\DeclareSizeFunction{sgenb}{\genb@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\genb@sfcnt{%
   \edef\mandatory@arg{\mandatory@arg\expandafter\genb@x\f@size..\@@}%
   \empty@sfcnt}
⟨/2ekernel | def2 | autoload⟩
```

`\genb@x` `\genb@y` The auxiliary macros `\genb@x` and `\genb@y` are used to convert the `\f@size` into centipoints.

```
⟨*2ekernel | def2 | autoload⟩
\def\genb@x#1.#2.#3\@@{\two@digits{#1}\genb@y#200\@@}
\def\genb@y#1#2#3\@@{#1#2}
⟨/2ekernel | def2 | autoload⟩
```

`\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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{sub}{\sub@sfcnt\@font@warning}
\DeclareSizeFunction{ssub}{\sub@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\sub@sfcnt#1{%
    \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.

```
    \begingroup
     \expandafter\split@name\mandatory@arg/\@nil
     \try@load@fontshape
    \endgroup
```

Then we record the current `\f@size` since it may get clobbered.

```
    \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`.

```
    \expandafter
    \ifx\csname\mandatory@arg\endcsname\relax
      \errmessage{No\space declaration\space for\space
                  shape\space \mandatory@arg}%
      \error@fontshape
    \else
```

Otherwise we warn the user about the substitution taking place.

```
      #1{Font\space shape\space `\curr@fontshape'\space in\space
         size\space <\f@size>\space not\space available\MessageBreak
         Font\space shape\space `\mandatory@arg'\space tried\space
         instead}%
      \expandafter\split@name\mandatory@arg/\@nil
    \fi
```

Then we restart the font specification scan by calling `\get@external@font`.

```
    \edef\f@size{\f@user@size}%
    \get@external@font
```

Finally `\do@subst@correction` is called to get the font name right.

```
    \do@subst@correction
}
⟨/2ekernel | def2 | autoload⟩
```

`\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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{subf}{\subf@sfcnt\@font@warning}
\DeclareSizeFunction{ssubf}{\subf@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\subf@sfcnt#1{%
      #1{Font\space shape\space `\curr@fontshape'\space in\space
         size\space \f@size\space not\space available\MessageBreak
        external\space font\space `\mandatory@arg'\space used}%
      \empty@sfcnt#1%
      }
⟨/2ekernel | def2 | autoload⟩
```

`\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.

```
⟨*2ekernel | autoload⟩
\DeclareSizeFunction{fixed}{\fixed@sfcnt\@font@warning}
\DeclareSizeFunction{sfixed}{\fixed@sfcnt\@font@info}
⟨/2ekernel | autoload⟩
```

```
⟨*2ekernel | def2 | autoload⟩
\def\fixed@sfcnt#1{%
  \ifx\optional@arg\@empty
    \let\external@font\mandatory@arg
  \else
    \edef\external@font{\mandatory@arg\space at\optional@arg pt}%
  \fi
  #1{External\space font\space ‘\external@font’\space loaded\space
     for\space size\MessageBreak
     <\f@size>}%
}
⟨/2ekernel | def2 | autoload⟩
```

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.

34 Compatibility code for NFSS release 1

There have been a couple of commands which became obsolete with NFSS2. In the past they have been still part of the kernel code to make it possible to process old packages using those commands but since they take up valuable space we decided to remove them and instead auto-load their definitions if they are actually encountered in some file.

Thus the following code doesn’t really belong to this file but I put it here for the moment until finally a documented version of `ltfss.dtx` is available.

[auto-loading not activated]

`\new@fontshape`
`\subst@fontshape`
`\extra@def`
`\default@mextra`
`\define@mathalphabet`
`\define@mathgroup`

These macros are the interfaces in NFSS1 which shouldn’t be used any longer. We all define them to call the macro `\scan@fontshape` which is an internal macro that loads the real definitions and then to execute themselves again. Once this auto-loading has happened they have the definition shown below and thus execute their real code directly.

```
⟨*autoload⟩
\def\new@fontshape{\scan@fontshape\new@fontshape}
\def\subst@fontshape{\scan@fontshape\subst@fontshape}
\def\extra@def{\scan@fontshape\extra@def}
\def\default@mextra{\scan@fontshape\default@mextra}
\def\define@mathalphabet{\scan@fontshape\define@mathalphabet}
\def\define@mathgroup{\scan@fontshape\define@mathgroup}
```

`\scan@fontshape`

Here is the kernel definition for `\scan@fontshape` which loads the actual definitions from the file `nfsscmp.def`.

```
\def\scan@fontshape{\input{nfsscmp.def}}
```

The following definitions are now placed into the auto-load file.

Since we don’t know when this file will be read in we need to provide ourselves with standard `\catcode` settings. This is done by placing all definitions in a group and calling `\nfss@catcodes`. But this macro will also disable spaces which isn’t very appropriate for the following code because it contains a lot of helper messages. Therefore we change this back.

```
\begingroup
\nfss@catcodes
\catcode`\ =10\relax
⟨/autoload⟩
⟨*compat⟩
```

`\new@fontshape`

The interface is now `\DeclareFontShape`.

```
\gdef\new@fontshape#1#2#3#4{%
     \warn@rel@i\new@fontshape\DeclareFontShape
     \expandafter\scan@fontshape\@gobble#4<\@nil><<%
     \DeclareFontShape U{#1}{#2}{#3}\reserved@f}
\@onlypreamble\new@fontshape
```

`\warn@rel@i` The warning message used above.

```
\gdef\warn@rel@i#1#2{%
 \@font@warning{***  NFSS release 1 command
               \noexpand#1found\MessageBreak
   ***  Update by using release 2 command
        \string#2.\MessageBreak
   ***  Recovery is probably possible}%
}
\@onlypreamble\warn@rel@i
```

`\scan@fontshape` This will scan the old font shape definition syntax.

```
\gdef\scan@fontshape{%
  \let\reserved@f\@empty
  \let\reserved@e\@empty %        holds last info
  \scan@@fontshape
}
\@onlypreamble\scan@fontshape
```

`\scan@@fontshape`

```
\gdef\scan@@fontshape#1>#2#3<{%
  \ifx\@nil#1%
    \edef\reserved@f{\reserved@f\reserved@e}%
  \else
    \def\reserved@b{#1}%       nick names
    \def\reserved@c{#3}%
    \in@{ at}{#3}%
    \ifin@
      \in@{pt}{#3}%  not a proof but a good chance
      \ifin@
```

We grap also everything after pt and discard it if people have forgotten to place a percent sign there.

```
        \def\reserved@a##1 at##2pt##3\@nil{%
           \def\reserved@b{##2}%
           \def\reserved@c{##1}%
           }%
        \reserved@a#3\@nil
      \fi
    \fi
    \ifnum 0<0#2
      \edef\reserved@d{subf*\reserved@c}%
      \ifcase #2\or
      \or
      \else
        \errmessage{*** What's this? NFSS release 0? ***}%
      \fi
    \else
      \edef\reserved@d{#2\reserved@c}%
    \fi
    \ifx\reserved@d\reserved@e
      \edef\reserved@f{\reserved@f<\reserved@b>}%
    \else
      \edef\reserved@f{\reserved@f\reserved@e<\reserved@b>}%add old info
      \let\reserved@e\reserved@d
    \fi
    \expandafter\scan@@fontshape
  \fi
}
\@onlypreamble\scan@@fontshape
```

`\subst@fontshape` This is now also handled by the extend syntax of `\DeclareFontShape`.

```
\gdef\subst@fontshape#1#2#3#4#5#6{%
```

```
     \warn@rel@i\subst@fontshape\DeclareFontShape
     \DeclareFontShape{U}{#1}{#2}{#3}{<->sub*#4/#5/#6}{}}
\@onlypreamble\subst@fontshape
```

`\extra@def` This was replaced by `\DeclareFontFamily`.

```
\gdef\extra@def#1#2#3{%
     \warn@rel@i\extra@def\DeclareFontFamily
     \DeclareFontFamily{U}{#1}{}%
}
\@onlypreamble\extra@def
```

`\default@mextra` The new name is `\DeclareFontEncodingDefaults` but in this case we don't feel comfortable with this either.

```
\gdef\default@mextra{%
  \warn@rel@i\default@mextra\DeclareFontEncodingDefaults
```

We pick up the argument to `\default@mextra` implicitly as the second argument of `\DeclareFontEncodingDefaults`.

```
  \DeclareFontEncodingDefaults\relax
}
\@onlypreamble\default@mextra
```

`\preload@sizes` The new interface is `\DeclarePreloadSizes`.

```
\gdef\preload@sizes{%
     \warn@rel@i\preload@sizes\DeclarePreloadSizes
     \DeclarePreloadSizes U%
}
\@onlypreamble\preload@sizes
```

`\err@rel@i` This macro is used in cases where emulation with NFSS2 features is not really possible.

```
\gdef\err@rel@i#1#2{%
  \@latex@error{***  NFSS release 1 command \noexpand#1found%
          ^^J***  Recovery not possible. Use \string#2}%
       {The new release of NFSS doesn't support the
        \noexpand#1command^^Jany longer.
        Please upgrade your file to the syntax of NFSS
        release 2^^Jusing the \noexpand#2command.}%
```

Let's die.

```
  \batchmode\input.\relax
}
\@onlypreamble\err@rel@i
```

`\newmathalphabet`
`\newmathalphabet@@`
`\newmathalphabet@@@`

`\newmathalphabet` is the old form.

```
\gdef\newmathalphabet{%
  \if@no@font@opt
    \@latex@error{*** NFSS release 1 command
                    \noexpand\newmathalphabet found%
     ^^J \space*** Automatic recovery not possible.%
     ^^J \space*** TYPE H for Help%
              }%
       {Please look at the file usrguide.tex for hints on
        how to resolve this problem.}%
  \else
    \warn@rel@i\newmathalphabet\DeclareMathAlphabet
  \fi
  \@ifstar\newmathalphabet@@@
          \newmathalphabet@@}
\gdef\newmathalphabet@@#1{\DeclareMathAlphabet#1{U}{}{}{}}
\gdef\newmathalphabet@@@#1#2#3#4{%
       \DeclareMathAlphabet{#1}{U}{#2}{#3}{#4}}
```

```
\@onlypreamble\newmathalphabet
\@onlypreamble\newmathalphabet@@
\@onlypreamble\newmathalphabet@@@
```

`\if@no@font@opt`
`\@no@font@optfalse`

```
\global\let\if@no@font@opt\iftrue
\gdef\@no@font@optfalse{\let\if@no@font@opt\iffalse}
```

`\define@mathalphabet` This is a case where dying is best.

```
\gdef\define@mathalphabet{%
      \err@rel@i\define@mathalphabet\DeclareMathAlphabet
}
\@onlypreamble\define@mathalphabet
```

`\define@mathgroup` And here is another one

```
\gdef\define@mathgroup{%
      \err@rel@i\define@mathgroup\DeclareSymbolFont
}
\@onlypreamble\define@mathgroup
⟨/compat⟩
```

`\addtoversion` `\addtoversion` is the old form.

```
\def\addtoversion#1#2{%
  \warn@rel@i\addtoversion\SetMathAlphabet
  \SetMathAlphabet#2{#1}{U}}
\@onlypreamble\addtoversion
```

That finishes the definitions for the old interfaces — but first we better finish the group.

```
⟨*autoload⟩
\endgroup
⟨/autoload⟩
```

File r

ltfssdcl.dtx

This file contains the main implementation of the font selection scheme commands. See other parts of the LaTeX distribution, or *The LaTeX 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.

35 Interface Commands

`\in@`
`\ifin@`

`\@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).

```
⟨*2ekernel | autoload⟩
\def\in@#1#2%
 {%
   \begingroup
     \def\in@@##1#1{}%
     \toks@\expandafter{\in@@#2{}{}#1}%
     \edef\in@@{\the\toks@}%
   \expandafter\endgroup
   \ifx\in@@\@empty
     \in@false
   \else
     \in@true
   \fi
 }
\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`$\langle version_1\rangle$`\version@elt`$\langle version_2\rangle\ldots$
  `\version@elt`$\langle version_n\rangle$

- the list of all math alphabet identifiers. Here every entry has the form:
  `\group@elt`⟨*math group number*⟩
  `{{`⟨*default family*⟩`}{`⟨*default series*⟩`}{`⟨*default shape*⟩`}}`.

- Each defined math alphabet identifier holds a list containing Information about the *versions* for which it is defined. This list has a more complicated structure: it looks as follows:

  `\set@alpha`⟨*the alphabet identifier itself*⟩
     `\reserved@c`⟨*math version*⟩⟨*font info*⟩
    ...
  `\@nil`

where ⟨*font info*⟩ is either `\reserved@e` (if the combination is not defined yet) or

`{{`⟨*family*⟩`}{`⟨*series*⟩`}{`⟨*shape*⟩`}}`

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

```
\let\version@list=\@empty
\@onlypreamble\version@list
```

`\version@elt`

```
\let\version@elt\relax
\@onlypreamble\version@elt
```

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

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

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

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

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

```
%    \@latex@error{Math version
%               ‘\expandafter\@gobblefour\string#1’
%               already defined}\@eha
```

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

```
%  \else
%     \global\expandafter\newcount\csname c@\expandafter
%                                  \@gobble\string#1\endcsname
%     \global\csname c@\expandafter
%                                  \@gobble\string#1\endcsname\@ne
%     \def\version@elt{\noexpand\version@elt\noexpand}%
%     \edef\version@list{\version@list\version@elt#1}%
```

Then we prepare to enter the new version into all math alphabet identifier lists. Remember that these lists use `\reserved@c` as delimiter, and that there appears the control sequence `\reserved@e` that must not be expanded. Therefore we take suitable precautions.

```
%     \def\reserved@c{\noexpand\reserved@c\noexpand}%
%     \let\reserved@e\relax
```

We will now go through the `\alpha@list` to process every ⟨*math alphabet identifier*⟩ 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).

```
%     \def\group@elt##1##2##3{%
```

The first of these arguments is the ⟨*math alphabet identifier*⟩. 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.

```
%          \edef##1{\expandafter\remove@nil##1%
%                   \reserved@c
%                   #1%
%                   \reserved@e
%                   \noexpand\@nil}}%
```

Finally we call `\alpha@list` which will now execute the macro `\group@elt` once for every defined ⟨*math alphabet identifier*⟩. And that's all for now.

```
%     \alpha@list
%  \fi}
```

`\alpha@list` As we explained above every entry in `\alpha@list` has the form

`\alpha@elt`
⟨*alphabet identifier*⟩⟨*internal group number*⟩⟨*default font assignments*⟩...

We initialize it to `\@empty`.

```
\let\alpha@list\@empty
\@onlypreamble\alpha@list
```

`\alpha@elt`

```
\let\alpha@elt\relax
\@onlypreamble\alpha@elt
```

`\newgroup` Start the group (fam) allocation at 0. (Doesn't belong here.)

```
\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.

```
\def\select@group#1#2#3#4{%
 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
 {%
 \ifmmode
  \ifnum\csname c@mv@\math@version\endcsname<\sixt@@n
     \begingroup
       \escapechar\m@ne
       \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
       \globaldefs\@ne  \math@fonts
     \endgroup
     \init@restore@version
     \xdef#1{\noexpand\use@mathgroup\noexpand#2%
             {\number\csname c@mv@\math@version\endcsname}}%
     \global\advance\csname c@mv@\math@version\endcsname\@ne
   \else
     \let#1\relax
     \@latex@error{Too many math alphabets used in
                   version \math@version}%
        \@eha
   \fi
 \else \expandafter\non@alpherr\fi
 #1{#4}%
 }%
}
\@onlypreamble\restore@mathversion
```

`\init@restore@version`

```
\def\init@restore@version{%
```

```
        \global\let\init@restore@version\relax
        \xdef\restore@mathversion
             {\expandafter\noexpand\csname mv@\math@version\endcsname
              \global\csname c@mv@\math@version\endcsname
              \number\csname c@mv@\math@version\endcsname\relax}%
        \aftergroup\dorestore@version
}
\@onlypreamble\init@restore@version
```

`\non@alpherr`

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | autoerr⟩
\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.

```
    \string#1allowed only in math mode}\@ehd}
⟨/2ekernel | autoerr⟩
⟨autoload⟩\def\non@alpherr{\@autoerr\non@alpherr}
⟨*2ekernel | autoload⟩
```

`\dorestore@version`

```
\def\dorestore@version
 {\ifmmode
    \aftergroup\dorestore@version
  \else
    \gdef\init@restore@version{%
        \global\let\init@restore@version\relax
        \xdef\restore@mathversion
             {\expandafter\noexpand\csname mv@\math@version\endcsname
              \global\csname c@mv@\math@version\endcsname
              \number\csname c@mv@\math@version\endcsname\relax}%
        \aftergroup\dorestore@version
    }%
    \begingroup
      \let\getanddefine@fonts\@gobbletwo
      \restore@mathversion
    \endgroup
  \fi}%
\@onlypreamble\dorestore@version
```

`\document@select@group` We surround `\select@group` with braces so that functions using it can be used directly after `_` or `^`.

```
\def\document@select@group#1#2#3#4{%
 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
 {%
 \ifmmode
   \ifnum\csname c@mv@\math@version\endcsname<\sixt@@n
     \begingroup
       \escapechar\m@ne
       \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
       \globaldefs\@ne  \math@fonts
     \endgroup
     \expandafter\extract@alph@from@version
         \csname mv@\math@version\expandafter\endcsname
         \expandafter{\number\csname
                       c@mv@\math@version\endcsname}%
          #1%
     \global\advance\csname c@mv@\math@version\endcsname\@ne
   \else
     \let#1\relax
     \@latex@error{Too many math alphabets used
```

```
                   in version \math@version}%
        \@eha
  \fi
 \else \expandafter\non@alpherr\fi
 #1{#4}%
 }%
}
```

`\process@table`

```
\def\process@table{%
    \def\cdp@elt##1##2##3##4{%
        \@font@info{Checking defaults for
                  ##1/##2/##3/##4}%
        \expandafter
        \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.

```
          \begingroup
           \def\f@encoding{##1}\def\f@family{##2}%
           \try@load@fontshape
          \endgroup
        \fi
        \expandafter
        \ifx\csname##1/##2/##3/##4\endcsname\relax
             \@latex@error{This NFSS system isn't set up properly}%
                       {For encoding scheme ##1 the defaults
                        ##2/##3/##4 do not form a valid font shape}%
        \else
             \@font@info{... okay}%
        \fi}%
    \cdp@list
```

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

```
    \begingroup
       \escapechar\m@ne
       \error@fontshape
       \expandafter\ifx\csname \curr@fontshape\endcsname\relax
          \begingroup
            \try@load@fontshape
           \endgroup
       \fi
       \expandafter\ifx\csname \curr@fontshape\endcsname\relax
         \@latex@error{This NFSS system isn't set up properly}%
            {The system maintainer forgot to specify a suitable
             substitution
             font shape using the \noexpand\DeclareErrorFont
             command}%
       \fi
    \endgroup
```

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

```
    \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`.

```
    \fontencoding{\encodingdefault}%
    \fontfamily{\familydefault}%
    \fontseries{\seriesdefault}%
    \fontshape{\shapedefault}%
```

kill all macros not longer needed. we need to add many more!!!!!!

```
 \everyjob{}%
}
\@onlypreamble\process@table
```

```
%\@onlypreamble\set@mathradical
```

\DeclareMathVersion

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

\new@mathversion

```
\def\new@mathversion#1{%
  \expandafter\in@\expandafter#1\expandafter{\version@list}%
  \ifin@
    \@font@info{Redeclaring math version
               `\expandafter\@gobblefour\string#1'}%
  \else
    \global\expandafter\newcount\csname c@\expandafter
                                \@gobble\string#1\endcsname
    \def\version@elt{\noexpand\version@elt\noexpand}%
    \edef\version@list{\version@list\version@elt#1}%
  \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.

```
  \toks@{}%
  \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.

```
  \def\group@elt##1##2{%
       \advance\count@\@ne
       \addto@hook\toks@{\getanddefine@fonts##1##2}%
       }%
  \group@list
```

We set the counter for this math version to the number of math groups found in `\group@list`.

```
  \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 distinugish the case that an alphabet by default should produce an error in new versions.

```
  \def\alpha@elt##1##2##3{%
       \ifx##2\no@alphabet@error
         \toks@\expandafter{\the\toks@\install@mathalphabet##1%
             {\no@alphabet@error##1}}%
       \else
         \toks@\expandafter{\the\toks@\install@mathalphabet##1%
             {\select@group##1##2##3}}%
       \fi
          }%
  \alpha@list
```

Finally we define the math version to expand to the contents of `\toks@`.

```
  \xdef#1{\the\toks@}%
}
\@onlypreamble\new@mathversion
```

\DeclareSymbolFont

```
\def\DeclareSymbolFont#1#2#3#4#5{%
 \@tempswafalse
 \edef\reserved@b{#2}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
 \if@tempswa
   \@ifundefined{sym#1}{%
      \expandafter\new@mathgroup\csname sym#1\endcsname
      \expandafter\new@symbolfont\csname sym#1\endcsname
                        {#2}{#3}{#4}{#5}}%
     {%
      \@font@info{Redeclaring symbol font `#1'}%
```

Update the group list.

```
      \def\group@elt##1##2{%
           \noexpand\group@elt\noexpand##1%
           \expandafter\ifx\csname sym#1\endcsname##1%
             \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
           \else
               \noexpand##2%
           \fi}%
      \xdef\group@list{\group@list}%
```

Update the version list.

```
      \def\version@elt##1{%
          \expandafter
          \SetSymbolFont@\expandafter##1\csname#2/#3/#4/#5\expandafter
              \endcsname \csname sym#1\endcsname
          }%
      \version@list
     }%
  \else
    \@latex@error{Encoding scheme  `#2' unknown}\@eha
  \fi
  }
\@onlypreamble\DeclareSymbolFont
```

`\group@list`

```
\let\group@list\@empty
\@onlypreamble\group@list
```

`\group@elt`

```
\let\group@elt\relax
\@onlypreamble\group@elt
```

`\new@symbolfont`

```
\def\new@symbolfont#1#2#3#4#5{%
    \toks@\expandafter{\group@list}%
    \edef\group@list{\the\toks@\noexpand\group@elt\noexpand#1%
                     \expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
    \def\version@elt##1{\toks@\expandafter{##1}%
                   \edef##1{\the\toks@\noexpand\getanddefine@fonts
                   #1\expandafter\noexpand\csname#2/#3/#4/#5\endcsname}%
                  \global\advance\csname c@\expandafter
                                 \@gobble\string##1\endcsname\@ne
                 }%
    \version@list
}
\@onlypreamble\new@symbolfont
```

`\SetSymbolFont`

```
\def\SetSymbolFont#1#2#3#4#5#6{%
```

```
 \@tempswafalse
 \edef\reserved@b{#3}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
 \if@tempswa
  \expandafter\SetSymbolFont@
    \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
    \endcsname \csname sym#1\endcsname
 \else
  \@latex@error{Encoding scheme  `#3' unknown}\@eha
 \fi
}
\@onlypreamble\SetSymbolFont
```

\SetSymbolFont@

```
\def\SetSymbolFont@#1#2#3{%
  \expandafter\in@\expandafter#1\expandafter{\version@list}%
  \ifin@
    \expandafter\in@\expandafter#3\expandafter{\group@list}%
    \ifin@
      \begingroup
        \expandafter\get@cdp\string#2\@nil\reserved@a
        \toks@{}%
        \def\install@mathalphabet##1##2{%
             \addto@hook\toks@{\install@mathalphabet##1{##2}}%
            }%
        \def\getanddefine@fonts##1##2{%
          \ifnum##1=#3%
             \addto@hook\toks@{\getanddefine@fonts#3#2}%
             \expandafter\get@cdp\string##2\@nil\reserved@b
             \ifx\reserved@a\reserved@b\else
                \@font@info{Encoding `\reserved@b' has changed
                    to `\reserved@a' for symbol font\MessageBreak
                   `\expandafter\@gobblefour\string#3' in the
                    math version `\expandafter
                    \@gobblefour\string#1'}%
             \fi
             \@font@info{%
                Overwriting symbol font
                `\expandafter\@gobblefour\string#3' in
                 version `\expandafter
                \@gobblefour\string#1'\MessageBreak
                \@spaces \expandafter\@gobble\string##2 -->
                         \expandafter\@gobble\string#2}%
          \else
             \addto@hook\toks@{\getanddefine@fonts##1##2}%
          \fi}%
         #1%
         \xdef#1{\the\toks@}%
      \endgroup
    \else
       \@latex@error{Symbol font `\expandafter\@gobblefour\string#3'
                  not defined}\@eha
    \fi
  \else
    \@latex@error{Math version `\expandafter\@gobblefour\string#1'
       is not
       defined}{You probably mispelled the name of the math
       version.^^JOr you have to specify an additional package.}%
  \fi
}
\@onlypreamble\SetSymbolFont@
```

`\get@cdp`

```
\def\get@cdp#1#2/#3\@nil#4{\def#4{#2}}
\@onlypreamble\get@cdp
```

`\DeclareMathAlphabet`

```
\def\DeclareMathAlphabet#1#2#3#4#5{%
 \@tempswafalse
 \edef\reserved@b{#2}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
 \if@tempswa
   \expandafter\ifx
   \csname\expandafter\@gobble\string#1\endcsname
   \relax
      \new@mathalphabet#1{#2}{#3}{#4}{#5}%
   \else
```

Check if it is already a math alphabet.

```
     \edef\reserved@a{\noexpand\in@{\string\select@group}%
         {\expandafter\meaning\csname \expandafter
          \@gobble\string#1\space\endcsname}}%
     \reserved@a
     \ifin@
       \@font@info{Redeclaring math alphabet \string#1}%
       \def\version@elt##1{%
         \expandafter\SetMathAlphabet@\expandafter
            ##1\csname#2/#3/#4/#5\expandafter\endcsname

            \csname M@#2\expandafter\endcsname
            \csname \expandafter\@gobble\string#1\space\endcsname#1}%
       \version@list
     \else
```

Check if it is a math alphabet defined via `\DeclareSymbolFontAlphabet`.

```
       \edef\reserved@a{\noexpand\in@{\string\use@mathgroup}%
         {\expandafter\meaning\csname \expandafter
          \@gobble\string#1\space\endcsname}}%
       \reserved@a
       \ifin@
```

In that case overwriting is simple since there is nothing inserted in the math version macros.

```
         \@font@info{Redeclaring math alphabet \string#1}%
         \new@mathalphabet#1{#2}{#3}{#4}{#5}%
```

Otherwise panic.

```
       \else
         \@latex@error{Command ‘\string#1’ already defined}\@eha
       \fi
     \fi
  \fi
 \else
  \@latex@error{Encoding scheme  ‘#2’ unknown}\@eha
 \fi
  }
\@onlypreamble\DeclareMathAlphabet
```

`\new@mathalphabet`

```
\def\new@mathalphabet#1#2#3#4#5{%
    \toks@\expandafter{\alpha@list}%
    \edef#1{\expandafter\noexpand\csname \expandafter
            \@gobble\string#1\space\endcsname
            \if/#5/%
               \noexpand\no@alphabet@error
               \noexpand\no@alphabet@error
            \else
               \expandafter\noexpand\csname M@#2\endcsname
               \expandafter\noexpand\csname#2/#3/#4/#5\endcsname
            \fi
           }%
    \toks2\expandafter{#1}%
    \edef\alpha@list{\the\toks@\noexpand\alpha@elt\the\toks2}%
    \def\version@elt##1{\toks@\expandafter{##1}%
                   \edef##1{\the\toks@\install@mathalphabet
                            \expandafter\noexpand
                            \csname \expandafter\@gobble
                               \string#1\space\endcsname
                           {\if/#5/%
                             \noexpand\no@alphabet@error
                             \noexpand#1%
                            \else
                             \noexpand\select@group\the\toks2
                            \fi}}%
                 }%
    \version@list
    \expandafter\edef\csname \expandafter\@gobble
                \string#1\space\endcsname{\if/#5/%
              \noexpand\no@alphabet@error
              \noexpand#1%
            \else
              \noexpand\select@group\the\toks2
            \fi}%
    \edef#1{\noexpand\protect
            \expandafter\noexpand\csname \expandafter
            \@gobble\string#1\space\endcsname}%
}
\@onlypreamble\new@mathalphabet
```

\SetMathAlphabet

```
\def\SetMathAlphabet#1#2#3#4#5#6{%
 \@tempswafalse
 \edef\reserved@b{#3}%
 \def\cdp@elt##1##2##3##4{\def\reserved@c{##1}%
      \ifx\reserved@b\reserved@c \@tempswatrue\fi}%
 \cdp@list
 \if@tempswa
  \expandafter\SetMathAlphabet@
    \csname mv@#2\expandafter\endcsname\csname#3/#4/#5/#6\expandafter
    \endcsname \csname M@#3\expandafter\endcsname
    \csname \expandafter\@gobble\string#1\space\endcsname#1%
 \else
  \@latex@error{Encoding scheme  `#3' unknown}\@eha
 \fi
}
\@onlypreamble\SetMathAlphabet
```

\SetMathAlphabet@

```
\def\SetMathAlphabet@#1#2#3#4#5{%
  \expandafter\in@\expandafter#1\expandafter{\version@list}%
  \ifin@
    \expandafter\in@\expandafter#4\expandafter{\alpha@list}%
    \ifin@
      \begingroup
        \toks@{}%
        \def\getanddefine@fonts##1##2{%
             \addto@hook\toks@{\getanddefine@fonts##1##2}%
            }%
        \def\reserved@c##1##2##3##4{%               % for message below
            \expandafter\@gobble\string##4}%
        \def\install@mathalphabet##1##2{%
          \ifx##1#4%
             \addto@hook\toks@
                {\install@mathalphabet#4{\select@group#4#3#2}}%
             \@font@info{Overwriting math alphabet
                `\string#5' in version `\expandafter
                 \@gobblefour\string#1'\MessageBreak
                 \@spaces \reserved@c##2 -->
                        \expandafter\@gobble\string#2}%
          \else
             \addto@hook\toks@{\install@mathalphabet##1{##2}}%
          \fi
          }%
        #1%
        \xdef#1{\the\toks@}%
      \endgroup
    \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.

```
       \edef\reserved@a{%
         \noexpand\in@{\string\use@mathgroup}{\meaning#4}}%
       \reserved@a
       \ifin@
         \def\reserved@b##1\use@mathgroup##2##3{%
             \def\reserved@b{##3}\def\reserved@c{##2}}%
         \expandafter\reserved@b#4%
         \begingroup
           \def\install@mathalphabet##1##2{%
               \addto@hook\toks@{\install@mathalphabet##1{##2}}%
               }%
            \def\getanddefine@fonts##1##2{%
              \addto@hook\toks@{\getanddefine@fonts##1##2}%
              \ifnum##1=\reserved@b
                 \expandafter
                 \addto@hook\expandafter\toks@
                 \expandafter{\expandafter\install@mathalphabet
                 \expandafter#4\expandafter
                       {\expandafter\select@group\expandafter
                         #4\reserved@c##2}}%
              \fi
                     }%
           \def\version@elt##1{%
               \toks@{}%
               ##1%
               \xdef##1{\the\toks@}%
              }%
           \version@list
         \endgroup
```

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

```
         \expandafter\gdef\expandafter\alpha@list\expandafter
            {\alpha@list
              \alpha@elt #4\no@alphabet@error \no@alphabet@error}%
         \gdef#4{\no@alphabet@error #5}% fake things :-)
```

Then call the internal setting routine again:

```
         \SetMathAlphabet@{#1}{#2}{#3}#4#5%
```

```
       \else
         \@latex@error{Command `\string#5' not defined as a
                       math alphabet}%
            {Use \noexpand\DeclareMathAlphabet to define it.}%
       \fi
    \fi
  \else
    \@latex@error{Math version `\expandafter\@gobblefour\string#1'
       is not
       defined}{You probably mispelled the name of the math
       version.^^JOr you have to specify an additional package.}%
  \fi
}
\@onlypreamble\SetMathAlphabet@
```

`\DeclareMathAlphabet` could do with more checks like allowing single number in #4 lowercase in #4 etc

```
\def\DeclareMathAccent#1#2#3#4{%
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \begingroup
      \count\z@=#4\relax
      \count\tw@\count\z@
      \divide\count\z@\sixt@@n
      \count@\count\z@
      \multiply\count@\sixt@@n
      \advance\count\tw@-\count@
      \if\relax\noexpand#1% is command?
        \edef\reserved@a{\noexpand\in@{\string\mathaccent}{\meaning#1}}%
        \reserved@a
        \ifin@
          \expandafter\set@mathaccent
             \csname sym#3\endcsname#1#2%
             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \@font@info{Redeclaring math accent \string#1}%
        \else
          \expandafter\ifx
          \csname\expandafter\@gobble\string#1\endcsname
          \relax
            \expandafter\set@mathaccent
               \csname sym#3\endcsname#1#2%
               {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \else
            \@latex@error{Command `\string#1' already defined}\@eha
          \fi
        \fi
      \else
       \@latex@error{Not a command name: `\noexpand#1'}\@eha
      \fi
    \endgroup
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi
}
\@onlypreamble\DeclareMathAccent
```

`\set@mathaccent`

```
\def\set@mathaccent#1#2#3#4{%
  \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}}
\@onlypreamble\set@mathaccent
```

`\DeclareMathSymbol`

```
\def\DeclareMathSymbol#1#2#3#4{%
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \begingroup
      \count\z@=#4\relax
      \count\tw@\count\z@
      \divide\count\z@\sixt@@n
      \count@\count\z@
      \multiply\count@\sixt@@n
      \advance\count\tw@-\count@
      \if\relax\noexpand#1% is command?
        \edef\reserved@a{\noexpand\in@{\string\mathchar}{\meaning#1}}%
        \reserved@a
        \ifin@
          \expandafter\set@mathsymbol
             \csname sym#3\endcsname#1#2%
             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \@font@info{Redeclaring math symbol \string#1}%
        \else
            \expandafter\ifx
            \csname\expandafter\@gobble\string#1\endcsname
            \relax
            \expandafter\set@mathsymbol
               \csname sym#3\endcsname#1#2%
               {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \else
            \@latex@error{Command `\string#1' already defined}\@eha
          \fi
        \fi
      \else
        \expandafter\set@mathchar
          \csname sym#3\endcsname#1#2
          {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      \fi
    \endgroup
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi
}
\@onlypreamble\DeclareMathSymbol
```

\set@mathchar

```
\def\set@mathchar#1#2#3#4{%
  \global\mathcode`#2="\mathchar@type#3\hexnumber@#1#4\relax}
\@onlypreamble\set@mathchar
```

\set@mathsymbol

```
\def\set@mathsymbol#1#2#3#4{%
  \global\mathchardef#2"\mathchar@type#3\hexnumber@#1#4\relax}
\@onlypreamble\set@mathsymbol
```

```
%\def\mathsymbol#1#2#3{%
%  \@tempcnta=#3\relax
%  \@tempcntb\@tempcnta
%  \divide\@tempcnta\sixt@@n
%  \count@\@tempcnta
%  \multiply\count@\sixt@@n
%  \advance\@tempcntb-\count@
%  \mathchar"\mathchar@type#1\hexnumber@#2%
%             \hexnumber@\@tempcnta\hexnumber@\@tempcntb\relax}
%
```

```
%\def\DeclareMathAlphabetCharacter#1#2#3{%
%  \DeclareMathSymbol{#1}7{#2}{#3}}
```

`\DeclareMathDelimiter`

```
\def\DeclareMathDelimiter#1{%
  \if\relax\noexpand#1%
    \expandafter\@DeclareMathDelimiter
  \else
    \expandafter\@xxDeclareMathDelimiter
  \fi
  #1}
\@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.

```
\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!

```
   \begingroup
    \let\mathalpha\mathord
    \ifnum7=\mathchar@type{#2}%
      \endgroup
```

If this branch is taken we have old syntax (5 arguments).

```
      \expandafter\@firstofone
    \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.

```
      \endgroup
      \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.

```
      \expandafter\@firstoftwo
    \fi
    {\@xDeclareMathDelimiter#1}{#2}{#3}{#4}}
\@onlypreamble\@xxDeclareMathDelimiter
```

`\@DeclareMathDelimiter`

```
\def\@DeclareMathDelimiter#1#2#3#4#5#6{%
  \expandafter\in@\csname sym#3\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \expandafter\in@\csname sym#5\expandafter\endcsname
       \expandafter{\group@list}%
    \ifin@
      \begingroup
        \count\z@=#4\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \count\z@=#6\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \edef\reserved@a{\noexpand\in@{\string\delimiter}{\meaning#1}}%
        \reserved@a
        \ifin@
          \expandafter\set@mathdelimiter
             \csname sym#3\expandafter\endcsname
             \csname sym#5\endcsname#1#2%
             \reserved@c\reserved@d
          \@font@info{Redeclaring math delimiter \string#1}%
        \else
            \expandafter\ifx
            \csname\expandafter\@gobble\string#1\endcsname
            \relax
            \expandafter\set@mathdelimiter
              \csname sym#3\expandafter\endcsname
              \csname sym#5\endcsname#1#2%
              \reserved@c\reserved@d
          \else
            \@latex@error{Command `\string#1' already defined}\@eha
          \fi
        \fi
      \endgroup
    \else
      \@latex@error{Symbol font `#5' is not defined}\@eha
    \fi
  \else
    \@latex@error{Symbol font `#3' is not defined}\@eha
  \fi
}
\@onlypreamble\@DeclareMathDelimiter
```

\@xDeclareMathDelimiter

```
\def\@xDeclareMathDelimiter#1#2#3#4#5{%
  \expandafter\in@\csname sym#2\expandafter\endcsname
     \expandafter{\group@list}%
  \ifin@
    \expandafter\in@\csname sym#4\expandafter\endcsname
       \expandafter{\group@list}%
    \ifin@
      \begingroup
        \count\z@=#3\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@c{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
      %
        \count\z@=#5\relax
        \count\tw@\count\z@
        \divide\count\z@\sixt@@n
        \count@\count\z@
        \multiply\count@\sixt@@n
        \advance\count\tw@-\count@
        \edef\reserved@d{\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
        \expandafter\set@@mathdelimiter
           \csname sym#2\expandafter\endcsname\csname sym#4\endcsname#1%
           \reserved@c\reserved@d
      \endgroup
    \else
      \@latex@error{Symbol font `#4' is not defined}\@eha
    \fi
  \else
    \@latex@error{Symbol font `#2' is not defined}\@eha
  \fi
}
\@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 contructs involving `\abovewithdelims` will prematurely end (pr/1329)

```
\def\set@mathdelimiter#1#2#3#4#5#6{%
  \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
                                      \hexnumber@#2#6 }}
\@onlypreamble\set@mathdelimiter
```

`\set@@mathdelimiter`

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

`\DeclareMathRadical`

```
\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!

```
  \expandafter\ifx
       \csname\expandafter\@gobble\string#1\endcsname
       \relax
     \let#1\radical
  \fi
  \edef\reserved@a{\noexpand\in@{\string\radical}{\meaning#1}}%
  \reserved@a
  \ifin@
    \expandafter\in@\csname sym#2\expandafter\endcsname
       \expandafter{\group@list}%
    \ifin@
      \expandafter\in@\csname sym#4\expandafter\endcsname
         \expandafter{\group@list}%
      \ifin@
        \begingroup
          \count\z@=#3\relax
          \count\tw@\count\z@
          \divide\count\z@\sixt@@n
          \count@\count\z@
          \multiply\count@\sixt@@n
          \advance\count\tw@-\count@
          \edef\reserved@c{%
            \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
          \count\z@=#5\relax
          \count\tw@\count\z@
          \divide\count\z@\sixt@@n
          \count@\count\z@
          \multiply\count@\sixt@@n
          \advance\count\tw@-\count@
          \edef\reserved@d{%
            \hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
```

Coded inline instead of using `\set@mathradical`

```
%          \expandafter\set@mathradical
%             \csname sym#2\expandafter\endcsname
%             \csname sym#4\endcsname#1%
%             \reserved@c\reserved@d
          \xdef#1{\radical"\expandafter\hexnumber@
                                \csname sym#2\endcsname\reserved@c
                             \expandafter\hexnumber@
                                \csname sym#4\endcsname\reserved@d
                  \relax}%
        \endgroup
      \else
        \@latex@error{Symbol font `#4' is not defined}\@eha
      \fi
    \else
      \@latex@error{Symbol font `#2' is not defined}\@eha
    \fi
  \else
    \@latex@error{Command `\string#1' already defined}\@eha
  \fi
}
\@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

```
\let\mathalpha\relax
```

`\mathchar@type`

```
\def\mathchar@type#1{%
  \ifodd 2#11 #1\else             % is this non-negative number?
    \ifx#1\mathord 0\else
     \ifx#1\mathop   1\else
       \ifx#1\mathbin 2\else
         \ifx#1\mathrel 3\else
           \ifx#1\mathopen 4\else
             \ifx#1\mathclose 5\else
               \ifx#1\mathpunct 6\else
                   7%             % anything else is variable ord
               \fi
             \fi
           \fi
         \fi
       \fi
     \fi
    \fi
  \fi}
\@onlypreamble\mathchar@type
```

`\DeclareSymbolFontAlphabet`

```
\def\DeclareSymbolFontAlphabet#1#2{%
   \expandafter\DeclareSymbolFontAlphabet@
     \csname \expandafter\@gobble\string#1\space\endcsname{#2}#1}
\@onlypreamble\DeclareSymbolFontAlphabet
```

`\DeclareSymbolFontAlphabet@`

```
\def\DeclareSymbolFontAlphabet@#1#2#3{%
```

We use the switch `\if@tempswa` to decide if we can declare this symbol font alphabet.

```
    \@tempswatrue
```

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

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

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

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

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

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

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

```
    \def\version@elt##1{%
        \begingroup
          \toks@{}%
          \def\getanddefine@fonts####1####2{%
             \addto@hook\toks@{\getanddefine@fonts####1####2}}%
          \def\install@mathalphabet####1####2{%
             \ifx####1#1\else
               \addto@hook\toks@{\install@mathalphabet
                                  ####1{####2}}\fi}%
          ##1%
          \xdef##1{\the\toks@}%
        \endgroup
        }%
    \version@list
  \else
```

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

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

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

```
    \else
      \edef\reserved@a{%
        \noexpand\in@{\string\use@mathgroup}{\meaning#1}}%
      \reserved@a
      \ifin@
        \@font@info{Redeclaring math alphabet \string#3}%
      \else
```

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

```
        \@tempswafalse
        \@latex@error{Command `\string#3' already defined}\@eha
      \fi
    \fi
  \fi
 \else
```

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

```
   \@tempswafalse
   \@latex@error{Unknown symbol font `#2'}\@eha
 \fi
 \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.

```
    \def\group@elt##1##2{%
       \expandafter\ifx\csname sym#2\endcsname##1%
       \expandafter\reserved@a\string##2\@nil
       \fi}%
    \def\reserved@a##1##2/##3\@nil{%
       \def\reserved@a{##2}}%
    \group@list
    \toks@{\relax\ifmmode \else \non@alpherr#1\fi}%
    \edef#1{\the\toks@
          \noexpand\use@mathgroup
          \expandafter\noexpand\csname M@\reserved@a\endcsname
          \csname sym#2\endcsname}%
    \def#3{\protect#1}%
  \fi
}
\@onlypreamble\DeclareSymbolFontAlphabet@
⟨/2ekernel | autoload⟩
```

File s

ltfssini.dtx

This file contains the top level LaTeX interface to the font selection scheme commands. See other parts of the LaTeX distribution, or *The LaTeX Companion* for higher level documentation of these commands.

36 NFSS Initialisation

Finally, there are six commands that are to be used in LaTeX and that we will therefore protect against expansion at the wrong point: `\fontfamily`, `\fontseries`, `\fontshape`, `\fontsize`, `\selectfont`, and `\mathversion`.

36.1 Providing math *versions*

LaTeX provides two *versions*. We call them normal and bold, respectively.

```
\DeclareMathVersion{normal}
\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*:

```
\DeclareRobustCommand\rmfamily
        {\not@math@alphabet\rmfamily\mathrm
         \fontfamily\rmdefault\selectfont}
\DeclareRobustCommand\sffamily
        {\not@math@alphabet\sffamily\mathsf
         \fontfamily\sfdefault\selectfont}
\DeclareRobustCommand\ttfamily
        {\not@math@alphabet\ttfamily\mathtt
         \fontfamily\ttdefault\selectfont}
```

Then the commands changing the *series*:

```
\DeclareRobustCommand\bfseries
        {\not@math@alphabet\bfseries\mathbf
         \fontseries\bfdefault\selectfont}
\DeclareRobustCommand\mdseries
        {\not@math@alphabet\mdseries\relax
         \fontseries\mddefault\selectfont}
\DeclareRobustCommand\upshape
        {\not@math@alphabet\upshape\relax
         \fontshape\updefault\selectfont}
```

Then the commands changing the *shape*:

```
\DeclareRobustCommand\slshape
        {\not@math@alphabet\slshape\relax
         \fontshape\sldefault\selectfont}
\DeclareRobustCommand\scshape
        {\not@math@alphabet\scshape\relax
         \fontshape\scdefault\selectfont}
\DeclareRobustCommand\itshape
        {\not@math@alphabet\itshape\mathit
         \fontshape\itdefault\selectfont}
```

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`.

```
\DeclareRobustCommand\em
        {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
                      \upshape \else \itshape \fi}
```

`\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`.

```
\def\not@math@alphabet#1#2{%
   \relax
   \ifmmode
     \@latex@error{Command \noexpand#1invalid in math mode}%
        {%
         Please
         \ifx#2\relax
            define a new math alphabet^^J%
            if you want to use a special font in math mode%
          \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 :-).

```
            use the math alphabet \noexpand#2instead of
            the #1command%
         \fi
         .
        }%
   \fi}
```

Finally we provide two abbreviations to switch to the LaTeX *versions.*

```
\def\boldmath{\@nomath\boldmath
              \mathversion{bold}}
\def\unboldmath{\@nomath\unboldmath
              \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`.

```
\def\math@version{normal}
```

36.2 Miscellaneous

`\newfont`
`\symbol`
We start by defining a few macros that are part of standard LaTeX's user interface. The use of these functions is not encouraged, but they will allow to process older documents without changes to the source.

```
\def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
\def\symbol#1{\char #1\relax}
```

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

```
\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`.

```
    \ifx\protect\@typeset@protect
      \let\@currsize#1%
    \fi
    \fontsize{#2}{#3}\selectfont}
```

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

```
⟨*compat⟩
\def\@setsize#1#2#3#4{\@setfontsize#1{#4}{#2}}
⟨/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.

```
\def\oldstylenums#1{%
   \begingroup
```

Provide spacing using the interword space of the current font.

```
    \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 LaTeX fonts even though they only contain substitutions).

```
    \usefont{OML}{\rmdefault}{\f@series}{it}%
    \mathgroup\symletters #1%
   \endgroup
}
```

`\hexnumber@` To set up LaTeX's special math character definitions we first provide a macro to generate hexadecimal numbers. It is a rather simple `\ifcase`.

```
\def\hexnumber@#1{\ifcase\number#1
 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or
 9\or A\or B\or C\or D\or E\or F\fi}
```

`\nfss@text` In it 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.

```
\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.

```
%\DeclareRobustCommand\copyright
%    {{\ooalign{\hfil
%     \raise.07ex\hbox{\mdseries\upshape c}\hfil\crcr
%     \mathhexbox20D}}}
```

`\normalfont`
`\reset@font`
`\p@reset@font`

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

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

```
\DeclareRobustCommand\normalfont
                 {\usefont\encodingdefault
                          \familydefault
                          \seriesdefault
                          \shapedefault
                  \relax}
\let\reset@font\normalfont
```

We left out the special LaTeX fonts which are not automatically included in the base version of the font selection since these fonts contain only a few characters which are also included in the AMS fonts so anybody who is using these fonts doesn't need them. But for compatibility reasons we will define these symbols.

```
\def\not@base#1{\@latex@error
  {Command \noexpand#1not provided in base LaTeX2e}%
  {Load the latexsym or the amsfonts package to
   define this symbol}}
\def\mho{\not@base\mho}
\def\Join{\not@base\Join}
\def\Box{\not@base\Box}
\def\Diamond{\not@base\Diamond}
\def\leadsto{\not@base\leadsto}
\def\sqsubset{\not@base\sqsubset}
\def\sqsupset{\not@base\sqsupset}
\def\lhd{\not@base\lhd}
\def\unlhd{\not@base\unlhd}
\def\rhd{\not@base\rhd}
\def\unrhd{\not@base\unrhd}
```

We now initialize all variables set by `\DeclareErrorFont`. These values are not really important since they will be overwritten later on by the definition in `fontdef.ltx`.

However, if `fontdef.cfg` is corrupted then at least a hopefully suitable error font is present.

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

We now load the customizable parts of NFSS.

```
\ifnum\inputlineno=\m@ne
```

Still using TEX2. need a configuration file to avoid setting the 8bit characters.

```
\InputIfFileExists{fonttext.cfg}
           {\typeout{====================================^^J%
                     ^^J%
                      Local config file fonttext.cfg used^^J%
                     ^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\typeout{!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
                     !^^J%
                     ! You MUST use a fonttext.cfg file!^^J%
                     ! As you are still using TeX2!!!!!^^J%
                     !^^J%
                     ! See the documentation file tex2.txt^^J%
                     !^^J%
                     !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!}%
                    \batchmode \@@end}
\else
```

With TEX3 can use the standard `ltx` file if no configuration file exists.

```
\InputIfFileExists{fonttext.cfg}
           {\typeout{====================================^^J%
                     ^^J%
                      Local config file fonttext.cfg used^^J%
                     ^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{fonttext.ltx}}
\fi
\let\@addtofilelist\@gobble
```

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

```
\InputIfFileExists{fontmath.cfg}
           {\typeout{====================================^^J%
                     ^^J%
                      Local config file fontmath.cfg used^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{fontmath.ltx}}
\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`.

```
\InputIfFileExists{preload.cfg}
           {\typeout{====================================^^J%
                     ^^J%
                      Local config file preload.cfg used^^J%
                     =====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{preload.ltx}}
\let\@addtofilelist\@gobble
```

`\@acci`
`\@accii`
`\@acciii`

We also save the values of some accents in `\@acci`, `\@accii` and `\@acciii` so they can be restored by a `minipage` inside a `tabbing` environment.

```
\let\@acci\' \let\@accii\` \let\@acciii\=
```

`\cal`
`\mit`

Here were the two old ⟨*alphabet identifiers*⟩.

File t

fontdef.dtx

37 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.

38 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 should create a copy of `fonttext.ltx` under the name `fonttext.cfg` and change this copy. If LaTeX 2_ε finds a file of this name it will use it, otherwise it uses the standard file which is `fontdef.ltx`.

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

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

> **Warning:** please note that we don't support customised LaTeX versions. Thus, before sending in a bug report please try your test file with a LaTeX format which is not customised and send in the log from that version (unless the problem goes away).

Please note: the following standard encodings have to be defined in all local variants of `font....cfg` to guarantee that all LaTeX installations behave in the same way.

`T1`	Cork TeX text encoding
`OT1`	old TeX text encoding
`U`	unknown encoding
`OML`	old TeX math letters encoding
`OMS`	old TeX math symbols encoding
`OMX`	old 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 TeX 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 TeX 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 LaTeX 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.

39 The `docstrip` modules

The following modules are used to direct `docstrip` in generating external files:

driver	produce a documentation driver file
text	produce the file `fonttext.ltx`
math	produce the file `fontmath.ltx`
cfgtext	produce a dummy `fonttext.cfg` file
cfgmath	produce a dummy `fontmath.cfg` file

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

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

40 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.

```
⟨*driver⟩
\documentclass{ltxdoc}
\GetFileInfo{fontdef.dtx}
\begin{document}
   \DocInput{fontdef.dtx}
\end{document}
⟨/driver⟩
```

41 The `fonttext.ltx` file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
⟨*text⟩
\typeout{=== Don't modify this file, use a .cfg file instead ===^^J}
```

41.1 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.

```
\input {omlenc.def}
\input  {t1enc.def}
\input {ot1enc.def}       % <- should come after T1 for speed
\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.

```
\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!

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

Then we define the default substitution for every encoding. This release of LaTeX 2ε 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.

```
\DeclareFontSubstitution{T1}{cmr}{m}{n}
\DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration, LaTeX 2ε 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. LaTeX 2ε 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 2ε 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.

```
\begingroup
\nfss@catcodes
\input  {t1cmr.fd}
\input  {ot1cmr.fd}
\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.)

```
\begingroup
\nfss@catcodes
\input {ot1cmss.fd}
\input {ot1cmtt.fd}
\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.

```
\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.

41.2 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`
`\sfdefault`
`\ttdefault`

The following three definitions set up the meaning for `\rmfamily`, `\sffamily`, and `\ttfamily`.

```
\newcommand\rmdefault{cmr}
\newcommand\sfdefault{cmss}
\newcommand\ttdefault{cmtt}
```

`\bfdefault`
`\mddefault`

Series changing commands are influenced by the following hooks.

```
\newcommand\bfdefault{bx}
\newcommand\mddefault{m}
```

`\itdefault`
`\sldefault`
`\scdefault`
`\updefault`

Shape changing commands use the following hooks.

```
\newcommand\itdefault{it}
\newcommand\sldefault{sl}
\newcommand\scdefault{sc}
\newcommand\updefault{n}
```

`\encodingdefault`
`\familydefault`
`\seriesdefault`
`\shapedefault`

Finally we have the hooks that describe the behaviour of the `\normalfont` command. To stay portable, the definition of `\encodingdefault` should *not* be changed and should match the setting above for `\fontencoding`. All other values can be set according to your taste.

```
\newcommand\encodingdefault{OT1}
\newcommand\familydefault{\rmdefault}
\newcommand\seriesdefault{\mddefault}
\newcommand\shapedefault{\updefault}
```

This finishes the low-level setup in `fonttext.ltx`.

```
⟨/text⟩
```

42 The `fontmath.ltx` file

The identification is done earlier on with a `\ProvidesFile` declaration.

```
⟨*math⟩
\typeout{=== Don't modify this file, use a .cfg file instead ===^^J}
```

42.1 The font encodings used

```
\DeclareFontEncoding{OML}{}{}
\DeclareFontEncoding{OMS}{}{}
\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.

```
\DeclareFontEncoding{U}{}{\noaccents@}
```

The encodings for math are next:

```
\DeclareFontSubstitution{OML}{cmm}{m}{it}
\DeclareFontSubstitution{OMS}{cmsy}{m}{n}
\DeclareFontSubstitution{OMX}{cmex}{m}{n}
\DeclareFontSubstitution{U}{cmr}{m}{n}
```

```
\begingroup
\nfss@catcodes
\input  {omlcmm.fd}
\input  {omscmsy.fd}
\input  {omxcmex.fd}
\input  {ucmr.fd}
\endgroup
```

42.1.1 Symbolfont and Alphabet declarations

We now define the basic symbol fonts used by LaTeX. These four symbol fonts must be defined by this file.

It is possible to make the symbol fonts point to other external fonts without losing the ability to process documents written at other sites, as long as one defines the same symbol font names with the same encodings, e.g. `operators` with `OT1` etc. If other encodings are used documents become non-portable. Such a change should therefore be done in a package file.

```
\DeclareSymbolFont{operators}   {OT1}{cmr} {m}{n}
\DeclareSymbolFont{letters}     {OML}{cmm} {m}{it}
\DeclareSymbolFont{symbols}     {OMS}{cmsy}{m}{n}
\DeclareSymbolFont{largesymbols}{OMX}{cmex}{m}{n}
```

```
\SetSymbolFont{operators}{bold}{OT1}{cmr} {bx}{n}
\SetSymbolFont{letters}  {bold}{OML}{cmm} {b}{it}
\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.

```
\DeclareSymbolFontAlphabet{\mathrm}    {operators}
\DeclareSymbolFontAlphabet{\mathnormal}{letters}
\DeclareSymbolFontAlphabet{\mathcal}   {symbols}
\DeclareMathAlphabet      {\mathbf}{OT1}{cmr}{bx}{n}
\DeclareMathAlphabet      {\mathsf}{OT1}{cmss}{m}{n}
\DeclareMathAlphabet      {\mathit}{OT1}{cmr}{m}{it}
\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.

```
\SetMathAlphabet\mathsf{bold}{OT1}{cmss}{bx}{n}
\SetMathAlphabet\mathit{bold}{OT1}{cmr}{bx}{it}
```

42.2 Math font sizes

The declarations below declare the text, script and scriptscript size to be used for each text font size.

All occurences 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).

```
\DeclareMathSizes{5}{5}{5}{5}
\DeclareMathSizes{6}{6}{5}{5}
\DeclareMathSizes{7}{7}{5}{5}
\DeclareMathSizes{8}{8}{6}{5}
\DeclareMathSizes{9}{9}{6}{5}
\DeclareMathSizes{\@xpt}{\@xpt}{7}{5}
\DeclareMathSizes{\@xipt}{\@xipt}{8}{6}
\DeclareMathSizes{\@xiipt}{\@xiipt}{8}{6}
\DeclareMathSizes{\@xivpt}{\@xivpt}{\@xpt}{7}
\DeclareMathSizes{\@xviipt}{\@xviipt}{\@xiipt}{\@xpt}
\DeclareMathSizes{\@xxpt}{\@xxpt}{\@xivpt}{\@xiipt}
\DeclareMathSizes{\@xxvpt}{\@xxvpt}{\@xxpt}{\@xviipt}
```

42.3 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.

42.3.1 The letters

```
\DeclareMathSymbol{a}{\mathalpha}{letters}{`a}
\DeclareMathSymbol{b}{\mathalpha}{letters}{`b}
\DeclareMathSymbol{c}{\mathalpha}{letters}{`c}
\DeclareMathSymbol{d}{\mathalpha}{letters}{`d}
\DeclareMathSymbol{e}{\mathalpha}{letters}{`e}
\DeclareMathSymbol{f}{\mathalpha}{letters}{`f}
\DeclareMathSymbol{g}{\mathalpha}{letters}{`g}
\DeclareMathSymbol{h}{\mathalpha}{letters}{`h}
\DeclareMathSymbol{i}{\mathalpha}{letters}{`i}
\DeclareMathSymbol{j}{\mathalpha}{letters}{`j}
\DeclareMathSymbol{k}{\mathalpha}{letters}{`k}
\DeclareMathSymbol{l}{\mathalpha}{letters}{`l}
\DeclareMathSymbol{m}{\mathalpha}{letters}{`m}
\DeclareMathSymbol{n}{\mathalpha}{letters}{`n}
\DeclareMathSymbol{o}{\mathalpha}{letters}{`o}
\DeclareMathSymbol{p}{\mathalpha}{letters}{`p}
\DeclareMathSymbol{q}{\mathalpha}{letters}{`q}
\DeclareMathSymbol{r}{\mathalpha}{letters}{`r}
\DeclareMathSymbol{s}{\mathalpha}{letters}{`s}
\DeclareMathSymbol{t}{\mathalpha}{letters}{`t}
\DeclareMathSymbol{u}{\mathalpha}{letters}{`u}
\DeclareMathSymbol{v}{\mathalpha}{letters}{`v}
\DeclareMathSymbol{w}{\mathalpha}{letters}{`w}
\DeclareMathSymbol{x}{\mathalpha}{letters}{`x}
\DeclareMathSymbol{y}{\mathalpha}{letters}{`y}
\DeclareMathSymbol{z}{\mathalpha}{letters}{`z}

\DeclareMathSymbol{A}{\mathalpha}{letters}{`A}
\DeclareMathSymbol{B}{\mathalpha}{letters}{`B}
\DeclareMathSymbol{C}{\mathalpha}{letters}{`C}
\DeclareMathSymbol{D}{\mathalpha}{letters}{`D}
\DeclareMathSymbol{E}{\mathalpha}{letters}{`E}
\DeclareMathSymbol{F}{\mathalpha}{letters}{`F}
\DeclareMathSymbol{G}{\mathalpha}{letters}{`G}
\DeclareMathSymbol{H}{\mathalpha}{letters}{`H}
\DeclareMathSymbol{I}{\mathalpha}{letters}{`I}
\DeclareMathSymbol{J}{\mathalpha}{letters}{`J}
```

```
\DeclareMathSymbol{K}{\mathalpha}{letters}{`K}
\DeclareMathSymbol{L}{\mathalpha}{letters}{`L}
\DeclareMathSymbol{M}{\mathalpha}{letters}{`M}
\DeclareMathSymbol{N}{\mathalpha}{letters}{`N}
\DeclareMathSymbol{O}{\mathalpha}{letters}{`O}
\DeclareMathSymbol{P}{\mathalpha}{letters}{`P}
\DeclareMathSymbol{Q}{\mathalpha}{letters}{`Q}
\DeclareMathSymbol{R}{\mathalpha}{letters}{`R}
\DeclareMathSymbol{S}{\mathalpha}{letters}{`S}
\DeclareMathSymbol{T}{\mathalpha}{letters}{`T}
\DeclareMathSymbol{U}{\mathalpha}{letters}{`U}
\DeclareMathSymbol{V}{\mathalpha}{letters}{`V}
\DeclareMathSymbol{W}{\mathalpha}{letters}{`W}
\DeclareMathSymbol{X}{\mathalpha}{letters}{`X}
\DeclareMathSymbol{Y}{\mathalpha}{letters}{`Y}
\DeclareMathSymbol{Z}{\mathalpha}{letters}{`Z}
```

42.3.2 The digits

```
\DeclareMathSymbol{0}{\mathalpha}{operators}{`0}
\DeclareMathSymbol{1}{\mathalpha}{operators}{`1}
\DeclareMathSymbol{2}{\mathalpha}{operators}{`2}
\DeclareMathSymbol{3}{\mathalpha}{operators}{`3}
\DeclareMathSymbol{4}{\mathalpha}{operators}{`4}
\DeclareMathSymbol{5}{\mathalpha}{operators}{`5}
\DeclareMathSymbol{6}{\mathalpha}{operators}{`6}
\DeclareMathSymbol{7}{\mathalpha}{operators}{`7}
\DeclareMathSymbol{8}{\mathalpha}{operators}{`8}
\DeclareMathSymbol{9}{\mathalpha}{operators}{`9}
```

42.3.3 Punctuation, brace, etc. keys

```
\DeclareMathSymbol{!}{\mathclose}{operators}{"21}
\DeclareMathSymbol{*}{\mathbin}{symbols}{"03} % \ast
\DeclareMathSymbol{+}{\mathbin}{operators}{"2B}
\DeclareMathSymbol{,}{\mathpunct}{letters}{"3B}
\DeclareMathSymbol{-}{\mathbin}{symbols}{"00}
\DeclareMathSymbol{.}{\mathord}{letters}{"3A}
\DeclareMathSymbol{:}{\mathrel}{operators}{"3A}
\DeclareMathSymbol{;}{\mathpunct}{operators}{"3B}
\DeclareMathSymbol{=}{\mathrel}{operators}{"3D}
\DeclareMathSymbol{?}{\mathclose}{operators}{"3F}
```

The following symbols are defined as delimiters below which automatically defines them as math symbols.

```
%\DeclareMathSymbol{(}{\mathopen}{operators}{"28}
%\DeclareMathSymbol{)}{\mathclose}{operators}{"29}
%\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
%\DeclareMathSymbol{[}{\mathopen}{operators}{"5B}
%\DeclareMathSymbol{]}{\mathclose}{operators}{"5D}
%\DeclareMathSymbol{|}{\mathord}{symbols}{"6A}
%\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
%\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
```

Should all of the following being activated by default? Probably not.

```
%\DeclareMathSymbol{`\{}{\mathopen}{symbols}{"66}
%\DeclareMathSymbol{`\}}{\mathclose}{symbols}{"67}
%\DeclareMathSymbol{`\\}{\mathord}{symbols}{"6E} % \backslash
\mathcode`\ ="8000 % \space
\mathcode`\'="8000 % ^\prime
\mathcode`\_="8000 % \_
```

42.3.4 Delimitercodes for characters

[to be completed]

Finally, IniTeX sets all `\delcode` values to -1, except ``\delcode`.=0``

```
\DeclareMathDelimiter{(}{\mathopen} {operators}{"28}{largesymbols}{"00}
\DeclareMathDelimiter{)}{\mathclose}{operators}{"29}{largesymbols}{"01}
\DeclareMathDelimiter{[}{\mathopen} {operators}{"5B}{largesymbols}{"02}
\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.

```
\DeclareMathDelimiter{<}{\mathopen}{symbols}{"68}{largesymbols}{"0A}
\DeclareMathDelimiter{>}{\mathclose}{symbols}{"69}{largesymbols}{"0B}
\DeclareMathSymbol{<}{\mathrel}{letters}{"3C}
\DeclareMathSymbol{>}{\mathrel}{letters}{"3E}
```

And here is another case where the non-delimiter version produces a glyph different from the delimiter version.

```
\DeclareMathDelimiter{/}{\mathord}{operators}{"2F}{largesymbols}{"0E}
\DeclareMathSymbol{/}{\mathord}{letters}{"3D}
```

```
\DeclareMathDelimiter{|}{\mathord}{symbols}{"6A}{largesymbols}{"0C}
```

```
\expandafter\DeclareMathDelimiter\@backslashchar
                          {\mathord}{symbols}{"6E}{largesymbols}{"0F}
```

N.B. `{` and `}` should NOT get delcodes; otherwise parameter grouping fails!

42.4 Symbols accessed via control sequences

42.4.1 Greek letters

```
\DeclareMathSymbol{\alpha}{\mathord}{letters}{"0B}
\DeclareMathSymbol{\beta}{\mathord}{letters}{"0C}
\DeclareMathSymbol{\gamma}{\mathord}{letters}{"0D}
\DeclareMathSymbol{\delta}{\mathord}{letters}{"0E}
\DeclareMathSymbol{\epsilon}{\mathord}{letters}{"0F}
\DeclareMathSymbol{\zeta}{\mathord}{letters}{"10}
\DeclareMathSymbol{\eta}{\mathord}{letters}{"11}
\DeclareMathSymbol{\theta}{\mathord}{letters}{"12}
\DeclareMathSymbol{\iota}{\mathord}{letters}{"13}
\DeclareMathSymbol{\kappa}{\mathord}{letters}{"14}
\DeclareMathSymbol{\lambda}{\mathord}{letters}{"15}
\DeclareMathSymbol{\mu}{\mathord}{letters}{"16}
\DeclareMathSymbol{\nu}{\mathord}{letters}{"17}
\DeclareMathSymbol{\xi}{\mathord}{letters}{"18}
\DeclareMathSymbol{\pi}{\mathord}{letters}{"19}
\DeclareMathSymbol{\rho}{\mathord}{letters}{"1A}
\DeclareMathSymbol{\sigma}{\mathord}{letters}{"1B}
\DeclareMathSymbol{\tau}{\mathord}{letters}{"1C}
\DeclareMathSymbol{\upsilon}{\mathord}{letters}{"1D}
\DeclareMathSymbol{\phi}{\mathord}{letters}{"1E}
\DeclareMathSymbol{\chi}{\mathord}{letters}{"1F}
\DeclareMathSymbol{\psi}{\mathord}{letters}{"20}
\DeclareMathSymbol{\omega}{\mathord}{letters}{"21}
\DeclareMathSymbol{\varepsilon}{\mathord}{letters}{"22}
\DeclareMathSymbol{\vartheta}{\mathord}{letters}{"23}
\DeclareMathSymbol{\varpi}{\mathord}{letters}{"24}
\DeclareMathSymbol{\varrho}{\mathord}{letters}{"25}
\DeclareMathSymbol{\varsigma}{\mathord}{letters}{"26}
\DeclareMathSymbol{\varphi}{\mathord}{letters}{"27}
\DeclareMathSymbol{\Gamma}{\mathalpha}{operators}{"00}
\DeclareMathSymbol{\Delta}{\mathalpha}{operators}{"01}
\DeclareMathSymbol{\Theta}{\mathalpha}{operators}{"02}
\DeclareMathSymbol{\Lambda}{\mathalpha}{operators}{"03}
\DeclareMathSymbol{\Xi}{\mathalpha}{operators}{"04}
```

```
\DeclareMathSymbol{\Pi}{\mathalpha}{operators}{"05}
\DeclareMathSymbol{\Sigma}{\mathalpha}{operators}{"06}
\DeclareMathSymbol{\Upsilon}{\mathalpha}{operators}{"07}
\DeclareMathSymbol{\Phi}{\mathalpha}{operators}{"08}
\DeclareMathSymbol{\Psi}{\mathalpha}{operators}{"09}
\DeclareMathSymbol{\Omega}{\mathalpha}{operators}{"0A}
```

42.4.2 Ordinary symbols

```
\DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
\def\hbar{{\mathchar’26\mkern-9muh}}
\DeclareMathSymbol{\imath}{\mathord}{letters}{"7B}
\DeclareMathSymbol{\jmath}{\mathord}{letters}{"7C}
\DeclareMathSymbol{\ell}{\mathord}{letters}{"60}
\DeclareMathSymbol{\wp}{\mathord}{letters}{"7D}
\DeclareMathSymbol{\Re}{\mathord}{symbols}{"3C}
\DeclareMathSymbol{\Im}{\mathord}{symbols}{"3D}
\DeclareMathSymbol{\partial}{\mathord}{letters}{"40}
\DeclareMathSymbol{\infty}{\mathord}{symbols}{"31}
\DeclareMathSymbol{\prime}{\mathord}{symbols}{"30}
\DeclareMathSymbol{\emptyset}{\mathord}{symbols}{"3B}
\DeclareMathSymbol{\nabla}{\mathord}{symbols}{"72}
\def\surd{{\mathchar"1270}}
\DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
\DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
\def\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crcr
      \not\mathrel{\mkern14mu}\crcr
      \noalign{\nointerlineskip}
      \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crcr}}}}
\DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
\DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
\DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
\DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}
    \let\lnot=\neg
\DeclareMathSymbol{\flat}{\mathord}{letters}{"5B}
\DeclareMathSymbol{\natural}{\mathord}{letters}{"5C}
\DeclareMathSymbol{\sharp}{\mathord}{letters}{"5D}
\DeclareMathSymbol{\clubsuit}{\mathord}{symbols}{"7C}
\DeclareMathSymbol{\diamondsuit}{\mathord}{symbols}{"7D}
\DeclareMathSymbol{\heartsuit}{\mathord}{symbols}{"7E}
\DeclareMathSymbol{\spadesuit}{\mathord}{symbols}{"7F}
```

42.4.3 Large Operators

```
\DeclareMathSymbol{\coprod}{\mathop}{largesymbols}{"60}
\DeclareMathSymbol{\bigvee}{\mathop}{largesymbols}{"57}
\DeclareMathSymbol{\bigwedge}{\mathop}{largesymbols}{"56}
\DeclareMathSymbol{\biguplus}{\mathop}{largesymbols}{"55}
\DeclareMathSymbol{\bigcap}{\mathop}{largesymbols}{"54}
\DeclareMathSymbol{\bigcup}{\mathop}{largesymbols}{"53}
\DeclareMathSymbol{\intop}{\mathop}{largesymbols}{"52}
    \def\int{\intop\nolimits}
\DeclareMathSymbol{\prod}{\mathop}{largesymbols}{"51}
\DeclareMathSymbol{\sum}{\mathop}{largesymbols}{"50}
\DeclareMathSymbol{\bigotimes}{\mathop}{largesymbols}{"4E}
\DeclareMathSymbol{\bigoplus}{\mathop}{largesymbols}{"4C}
\DeclareMathSymbol{\bigodot}{\mathop}{largesymbols}{"4A}
\DeclareMathSymbol{\ointop}{\mathop}{largesymbols}{"48}
    \def\oint{\ointop\nolimits}
\DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{"46}
\DeclareMathSymbol{\smallint}{\mathop}{symbols}{"73}
```

42.4.4 Binary symbols

```
\DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{"2F}
```

```
\DeclareMathSymbol{\triangleright}{\mathbin}{letters}{"2E}
\DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{"34}
\DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{"35}
   \let \varbigtriangledown \bigtriangledown
   \let \varbigtriangleup \bigtriangleup
```

These last two synonyms are needed because the stamryrd package redefines them as Operators.

```
\DeclareMathSymbol{\wedge}{\mathbin}{symbols}{"5E}
   \let\land=\wedge
\DeclareMathSymbol{\vee}{\mathbin}{symbols}{"5F}
   \let\lor=\vee
\DeclareMathSymbol{\cap}{\mathbin}{symbols}{"5C}
\DeclareMathSymbol{\cup}{\mathbin}{symbols}{"5B}
\DeclareMathSymbol{\ddagger}{\mathbin}{symbols}{"7A}
\DeclareMathSymbol{\dagger}{\mathbin}{symbols}{"79}
\DeclareMathSymbol{\sqcap}{\mathbin}{symbols}{"75}
\DeclareMathSymbol{\sqcup}{\mathbin}{symbols}{"74}
\DeclareMathSymbol{\uplus}{\mathbin}{symbols}{"5D}
\DeclareMathSymbol{\amalg}{\mathbin}{symbols}{"71}
\DeclareMathSymbol{\diamond}{\mathbin}{symbols}{"05}
\DeclareMathSymbol{\bullet}{\mathbin}{symbols}{"0F}
\DeclareMathSymbol{\wr}{\mathbin}{symbols}{"6F}
\DeclareMathSymbol{\div}{\mathbin}{symbols}{"04}
\DeclareMathSymbol{\odot}{\mathbin}{symbols}{"0C}
\DeclareMathSymbol{\oslash}{\mathbin}{symbols}{"0B}
\DeclareMathSymbol{\otimes}{\mathbin}{symbols}{"0A}
\DeclareMathSymbol{\ominus}{\mathbin}{symbols}{"09}
\DeclareMathSymbol{\oplus}{\mathbin}{symbols}{"08}
\DeclareMathSymbol{\mp}{\mathbin}{symbols}{"07}
\DeclareMathSymbol{\pm}{\mathbin}{symbols}{"06}
\DeclareMathSymbol{\circ}{\mathbin}{symbols}{"0E}
\DeclareMathSymbol{\bigcirc}{\mathbin}{symbols}{"0D}
\DeclareMathSymbol{\setminus}{\mathbin}{symbols}{"6E}
\DeclareMathSymbol{\cdot}{\mathbin}{symbols}{"01}
\DeclareMathSymbol{\ast}{\mathbin}{symbols}{"03}
\DeclareMathSymbol{\times}{\mathbin}{symbols}{"02}
\DeclareMathSymbol{\star}{\mathbin}{letters}{"3F}
```

42.4.5 Relations

```
\DeclareMathSymbol{\propto}{\mathrel}{symbols}{"2F}
\DeclareMathSymbol{\sqsubseteq}{\mathrel}{symbols}{"76}
\DeclareMathSymbol{\sqsupseteq}{\mathrel}{symbols}{"77}
\DeclareMathSymbol{\parallel}{\mathrel}{symbols}{"6B}
\DeclareMathSymbol{\mid}{\mathrel}{symbols}{"6A}
\DeclareMathSymbol{\dashv}{\mathrel}{symbols}{"61}
\DeclareMathSymbol{\vdash}{\mathrel}{symbols}{"60}
\DeclareMathSymbol{\nearrow}{\mathrel}{symbols}{"25}
\DeclareMathSymbol{\searrow}{\mathrel}{symbols}{"26}
\DeclareMathSymbol{\nwarrow}{\mathrel}{symbols}{"2D}
\DeclareMathSymbol{\swarrow}{\mathrel}{symbols}{"2E}
\DeclareMathSymbol{\Leftrightarrow}{\mathrel}{symbols}{"2C}
\DeclareMathSymbol{\Leftarrow}{\mathrel}{symbols}{"28}
\DeclareMathSymbol{\Rightarrow}{\mathrel}{symbols}{"29}
\def\neq{\not=} \let\ne=\neq
\DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
   \let\le=\leq
\DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}
   \let\ge=\geq
\DeclareMathSymbol{\succ}{\mathrel}{symbols}{"1F}
\DeclareMathSymbol{\prec}{\mathrel}{symbols}{"1E}
\DeclareMathSymbol{\approx}{\mathrel}{symbols}{"19}
\DeclareMathSymbol{\succeq}{\mathrel}{symbols}{"17}
```

```
\DeclareMathSymbol{\preceq}{\mathrel}{symbols}{"16}
\DeclareMathSymbol{\supset}{\mathrel}{symbols}{"1B}
\DeclareMathSymbol{\subset}{\mathrel}{symbols}{"1A}
\DeclareMathSymbol{\supseteq}{\mathrel}{symbols}{"13}
\DeclareMathSymbol{\subseteq}{\mathrel}{symbols}{"12}
\DeclareMathSymbol{\in}{\mathrel}{symbols}{"32}
\DeclareMathSymbol{\ni}{\mathrel}{symbols}{"33}
    \let\owns=\ni
\DeclareMathSymbol{\gg}{\mathrel}{symbols}{"1D}
\DeclareMathSymbol{\ll}{\mathrel}{symbols}{"1C}
\DeclareMathSymbol{\not}{\mathrel}{symbols}{"36}
\DeclareMathSymbol{\leftrightarrow}{\mathrel}{symbols}{"24}
\DeclareMathSymbol{\leftarrow}{\mathrel}{symbols}{"20}
   \let\gets=\leftarrow
\DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}
   \let\to=\rightarrow
\DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
   \def\mapsto{\mapstochar\rightarrow}
\DeclareMathSymbol{\sim}{\mathrel}{symbols}{"18}
\DeclareMathSymbol{\simeq}{\mathrel}{symbols}{"27}
\DeclareMathSymbol{\perp}{\mathrel}{symbols}{"3F}
\DeclareMathSymbol{\equiv}{\mathrel}{symbols}{"11}
\DeclareMathSymbol{\asymp}{\mathrel}{symbols}{"10}
\DeclareMathSymbol{\smile}{\mathrel}{letters}{"5E}
\DeclareMathSymbol{\frown}{\mathrel}{letters}{"5F}
\DeclareMathSymbol{\leftharpoonup}{\mathrel}{letters}{"28}
\DeclareMathSymbol{\leftharpoondown}{\mathrel}{letters}{"29}
\DeclareMathSymbol{\rightharpoonup}{\mathrel}{letters}{"2A}
\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.

```
\DeclareRobustCommand
  \cong{\mathrel{\mathpalette\@vereq\sim}} % congruence sign
\def\@vereq#1#2{\lower.5\p@\vbox{\lineskiplimit\maxdimen\lineskip-.5\p@
    \ialign{$\m@th#1\hfil##\hfil$\crcr#2\crcr=\crcr}}}
\DeclareRobustCommand
  \notin{\mathrel{\m@th\mathpalette\c@ncel\in}}
\def\c@ncel#1#2{\m@th\ooalign{$\hfil#1\mkern1mu/\hfil$\crcr$#1#2$}}
\DeclareRobustCommand
  \rightleftharpoons{\mathrel{\mathpalette\rlh@{}}}
\def\rlh@#1{\vcenter{\m@th\hbox{\ooalign{\raise2pt
          \hbox{$#1\rightharpoonup$}\crcr
        $#1\leftharpoondown$}}}}
\DeclareRobustCommand
  \doteq{\buildrel\textstyle.\over=}
```

42.4.6 Arrows

```
\DeclareRobustCommand
  \joinrel{\mathrel{\mkern-3mu}}
\DeclareRobustCommand
  \relbar{\mathrel{\smash-}}    % \smash, because -
                                % 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).

```
\DeclareRobustCommand
  \Relbar{\mathrel{=}}
\DeclareMathSymbol{\lhook}{\mathrel}{letters}{"2C}
   \def\hookrightarrow{\lhook\joinrel\rightarrow}
\DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
   \def\hookleftarrow{\leftarrow\joinrel\rhook}
\DeclareRobustCommand
  \bowtie{\mathrel\triangleright\joinrel\mathrel\triangleleft}
```

```
\DeclareRobustCommand
  \models{\mathrel{|}\joinrel\Relbar}
\DeclareRobustCommand
  \Longrightarrow{\Relbar\joinrel\Rightarrow}
```

LaTeX Change: `\longrightarrow` and `\longleftarrow` redefined to make then robust.

```
\DeclareRobustCommand\longrightarrow
     {\relbar\joinrel\rightarrow}
\DeclareRobustCommand\longleftarrow
     {\leftarrow\joinrel\relbar}
```

```
\DeclareRobustCommand
  \Longleftarrow{\Leftarrow\joinrel\Relbar}
\DeclareRobustCommand
  \longmapsto{\mapstochar\longrightarrow}
\DeclareRobustCommand
  \longleftrightarrow{\leftarrow\joinrel\rightarrow}
\DeclareRobustCommand
  \Longleftrightarrow{\Leftarrow\joinrel\Rightarrow}
\DeclareRobustCommand
  \iff{\;\Longleftrightarrow\;}
```

42.4.7 Punctuation symbols

```
\DeclareMathSymbol{\ldotp}{\mathpunct}{letters}{"3A}
\DeclareMathSymbol{\cdotp}{\mathpunct}{symbols}{"01}
\DeclareMathSymbol{\colon}{\mathpunct}{operators}{"3A}
```

This is commented out, since `\ldots` is now defined in ltoutenc.dtx.

```
%\def\@ldots{\mathinner{\ldotp\ldotp\ldotp}}
%\DeclareRobustCommand\ldots
%         {\relax\ifmmode\@ldots\else\mbox{$\m@th\@ldots\,$}\fi}
```

```
\DeclareRobustCommand
  \cdots{\mathinner{\cdotp\cdotp\cdotp}}
\DeclareRobustCommand
  \vdots{\vbox{\baselineskip4\p@ \lineskiplimit\z@
   \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
\DeclareRobustCommand
  \ddots{\mathinner{\mkern1mu\raise7\p@
   \vbox{\kern7\p@\hbox{.}}\mkern2mu
   \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu}}
```

42.4.8 Math accents

```
\DeclareMathAccent{\acute}{\mathalpha}{operators}{"13}
\DeclareMathAccent{\grave}{\mathalpha}{operators}{"12}
\DeclareMathAccent{\ddot}{\mathalpha}{operators}{"7F}
\DeclareMathAccent{\tilde}{\mathalpha}{operators}{"7E}
\DeclareMathAccent{\bar}{\mathalpha}{operators}{"16}
\DeclareMathAccent{\breve}{\mathalpha}{operators}{"15}
\DeclareMathAccent{\check}{\mathalpha}{operators}{"14}
\DeclareMathAccent{\hat}{\mathalpha}{operators}{"5E}
\DeclareMathAccent{\vec}{\mathord}{letters}{"7E}
\DeclareMathAccent{\dot}{\mathalpha}{operators}{"5F}
\DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
\DeclareMathAccent{\widehat}{\mathord}{largesymbols}{"62}
```

For some reason plain TEX 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:

```
\DeclareMathAccent{\mathring}{\mathalpha}{operators}{"17}
```

42.4.9 Radicals

```
\DeclareMathRadical{\sqrtsign}{symbols}{"70}{largesymbols}{"70}
```

42.4.10 Over and under something, etc

```
\def\overrightarrow#1{\vbox{\m@th\ialign{##\crcr
      \rightarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\def\overleftarrow#1{\vbox{\m@th\ialign{##\crcr
      \leftarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\def\overbrace#1{\mathop{\vbox{\m@th\ialign{##\crcr\noalign{\kern3\p@}%
      \downbracefill\crcr\noalign{\kern3\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}\limits}
\def\underbrace#1{\mathop{\vtop{\m@th\ialign{##\crcr
   $\hfil\displaystyle{#1}\hfil$\crcr
   \noalign{\kern3\p@\nointerlineskip}%
   \upbracefill\crcr\noalign{\kern3\p@}}}}\limits}
```

(quite a waste of tokens, IMHO — Frank)

```
\def\skew#1#2#3{{\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
    #2{\mkern-\muskip\z@{#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
```

```
\def\rightarrowfill{$\m@th\smash-\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
  \mkern-7mu\mathord\rightarrow$}
\def\leftarrowfill{$\m@th\mathord\leftarrow\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
  \mkern-7mu\smash-$}
\DeclareMathSymbol{\braceld}{\mathord}{largesymbols}{"7A}
\DeclareMathSymbol{\bracerd}{\mathord}{largesymbols}{"7B}
\DeclareMathSymbol{\bracelu}{\mathord}{largesymbols}{"7C}
\DeclareMathSymbol{\braceru}{\mathord}{largesymbols}{"7D}
\def\downbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
  \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru
  \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd$}
\def\upbracefill{$\m@th \setbox\z@\hbox{$\braceld$}%
  \bracelu\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\bracerd
  \braceld\leaders\vrule \@height\ht\z@ \@depth\z@\hfill\braceru$}
```

42.4.11 Delimiters

```
\DeclareMathDelimiter{\lmoustache}    % top from (, bottom from )
   {\mathopen}{largesymbols}{"7A}{largesymbols}{"40}
\DeclareMathDelimiter{\rmoustache}    % top from ), bottom from (
   {\mathclose}{largesymbols}{"7B}{largesymbols}{"41}
\DeclareMathDelimiter{\arrowvert}     % arrow without arrowheads
   {\mathord}{symbols}{"6A}{largesymbols}{"3C}
\DeclareMathDelimiter{\Arrowvert}     % double arrow without arrowheads
   {\mathord}{symbols}{"6B}{largesymbols}{"3D}
\DeclareMathDelimiter{\Vert}
   {\mathord}{symbols}{"6B}{largesymbols}{"0D}
\let\|=\Vert
\DeclareMathDelimiter{\vert}
   {\mathord}{symbols}{"6A}{largesymbols}{"0C}
\DeclareMathDelimiter{\uparrow}
   {\mathrel}{symbols}{"22}{largesymbols}{"78}
\DeclareMathDelimiter{\downarrow}
   {\mathrel}{symbols}{"23}{largesymbols}{"79}
\DeclareMathDelimiter{\updownarrow}
   {\mathrel}{symbols}{"6C}{largesymbols}{"3F}
```

```
\DeclareMathDelimiter{\Uparrow}
   {\mathrel}{symbols}{"2A}{largesymbols}{"7E}
\DeclareMathDelimiter{\Downarrow}
   {\mathrel}{symbols}{"2B}{largesymbols}{"7F}
\DeclareMathDelimiter{\Updownarrow}
   {\mathrel}{symbols}{"6D}{largesymbols}{"77}
\DeclareMathDelimiter{\backslash}    % for double coset G\backslash H
   {\mathord}{symbols}{"6E}{largesymbols}{"0F}
\DeclareMathDelimiter{\rangle}
   {\mathclose}{symbols}{"69}{largesymbols}{"0B}
\DeclareMathDelimiter{\langle}
   {\mathopen}{symbols}{"68}{largesymbols}{"0A}
\DeclareMathDelimiter{\rbrace}
   {\mathclose}{symbols}{"67}{largesymbols}{"09}
\DeclareMathDelimiter{\lbrace}
   {\mathopen}{symbols}{"66}{largesymbols}{"08}
\DeclareMathDelimiter{\rceil}
   {\mathclose}{symbols}{"65}{largesymbols}{"07}
\DeclareMathDelimiter{\lceil}
   {\mathopen}{symbols}{"64}{largesymbols}{"06}
\DeclareMathDelimiter{\rfloor}
   {\mathclose}{symbols}{"63}{largesymbols}{"05}
\DeclareMathDelimiter{\lfloor}
   {\mathopen}{symbols}{"62}{largesymbols}{"04}
```

`\lgroup`
`\rgroup`
`\bracevert`

There are three plain TEX delimiters which are not fully supported by NFSS, since they partly point into a bold cmr font. Allocating a full symbol font, just to have three delimiters seems a bit too much given the limited space available. For this reason only the extensible sizes are supported. If this is not desired one can use, without losing portability, define `\mathbf` and `\mathtt` as font symbol alphabet (setting up `cmr/bx/n` and `cmtt/m/n` as symbol fonts first) and modify the delimiter declarations to point with their small variant to those symbol fonts. (This is done in `oldlfont.dtx` so look there for examples.)

```
\DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
     {\mathopen}{largesymbols}{"3A}{largesymbols}{"3A}
\DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
     {\mathclose}{largesymbols}{"3B}{largesymbols}{"3B}
\DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
     {\mathord}{largesymbols}{"3E}{largesymbols}{"3E}
```

42.5 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`
`\mathsection`
`\mathdollar`
`\mathsterling`
`\mathunderscore`

These math symbols are not in plain TEX.

```
\DeclareMathSymbol{\mathparagraph}{\mathord}{symbols}{"7B}
\DeclareMathSymbol{\mathsection}{\mathord}{symbols}{"78}
\DeclareMathSymbol{\mathdollar}{\mathord}{operators}{"24}
```

```
\def\mathsterling{\mathit{\mathchar"7024}}
\def\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}
```

`\mathellipsis`

This is plain TEX's `\ldots`.

```
\def\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%
```

42.6 Other special functions and parameters

42.6.1 Biggggg

```
\def\big#1{{\hbox{$\left#1\vbox to8.5\p@{}\right.\n@space$}}}
```

```
\def\Big#1{{\hbox{$\left#1\vbox to11.5\p@{}\right.\n@space$}}}
\def\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{}\right.\n@space$}}}
\def\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{}\right.\n@space$}}}
\def\n@space{\nulldelimiterspace\z@ \m@th}
```

42.6.2 The log-like functions

`\operator@font` The `\operator@font` determines the symbol font used for log-like functions.

```
\def\operator@font{\mathgroup\symoperators}
```

42.6.3 Parameters

```
\thinmuskip=3mu
\medmuskip=4mu plus 2mu minus 4mu
\thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in `fontmath.ltx`.

```
⟨/math⟩
```

43 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.

```
⟨*cfgtext | cfgmath | cfgprel⟩
%%
%%
%%
%% Load the standard setup:
%%
⟨+cfgtext⟩\input{fonttext.ltx}
⟨+cfgmath⟩\input{fontmath.ltx}
⟨+cfgprel⟩\input{preload.ltx}
%%
%% Small changes could go here; see documentation in cfgguide.tex for
%% allowed modifications.
%%
%% In particular it is not allowed to misuse this configuration file
%% to modify internal LaTeX commands!
%%
%% If you use this file as the basis for configuration please change
%% the \ProvidesFile lines to clearly identify your modification, e.g.,
%%
⟨+cfgtext⟩%%  \ProvidesFile{fonttext.cfg}[2001/06/01
⟨+cfgmath⟩%%  \ProvidesFile{fonttext.cfg}[2001/06/01
⟨+cfgprel⟩%%  \ProvidesFile{preload.cfg}[2001/06/01
%%                            Customised local font setup]
%%
%%
⟨/cfgtext | cfgmath | cfgprel⟩
```

File u

preload.dtx

44 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by LaTeX 2ε). 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 `lfonts.tex`
preload.ltx	The standard selection of preloads
cmpreloa.xpt	preload of CM fonts for 10pt document size
cmpreloa.xip	preload of CM fonts for 11pt document size
cmpreloa.xii	preload of CM fonts for 12pt document size
dcpreloa.xpt	preload of DC fonts for 10pt size
dcpreloa.xip	preload of DC fonts for 11pt size
dcpreloa.xii	preload of DC fonts for 12pt size

These files are for installations that make use of Computer Modern fonts either old encoding (OT1) or Cork encoding (T1). The Computer Modern fonts with Cork encoding are known as DC-fonts.

Most important is `preload.ltx` which is used during format generation. You are *not* allowed to change this file.

45 Customization

You can customize the preloaded fonts in your LaTeX 2ε 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* LaTeX 2ε sytems.

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 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.

46 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 `lfonts.tex`
tex	produce preload.ltx

A typical DOCSTRIP command file would then have entries like:

```
\generateFile{preload.min}{t}{\from{preload.dtx}{preload,min}}
```

for generating preload files.

47 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.

```
⟨*driver⟩
\documentclass{ltxdoc}
%\OnlyDescription  % comment out for implementation details
\begin{document}
   \DocInput{preload.dtx}
\end{document}
⟨/driver⟩
```

48 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.

```
\expandafter\font\csname OMX/cmex/m/n/10\endcsname=cmex10\relax
\font\tenln  =line10   \font\tenlnw  =linew10\relax
\font\tencirc=lcircle10 \font\tencircw=lcirclew10\relax
```

The above fonts should not be touched but anything below this point here in the preload suggestions can be modified without any problems.

```
⟨-tex⟩%******************************************
⟨-tex⟩% Start any modification below this point **
⟨-tex⟩%******************************************
⟨-tex⟩
%%
%% Computer Modern Roman:
%%-----------------------
⟨*ori⟩
\DeclarePreloadSizes{OT1}{cmr}{m}{n}
         {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74,24.88}
\DeclarePreloadSizes{OT1}{cmr}{bx}{n}{9,10,10.95,12,14.4,17.28}
\DeclarePreloadSizes{OT1}{cmr}{m}{sl}{10,10.95,12}
\DeclarePreloadSizes{OT1}{cmr}{m}{it}{7,8,9,10,10.95,12}
⟨/ori⟩
⟨+xpt & cm⟩ \DeclarePreloadSizes{OT1}{cmr}{m}{n}{5,7,10}
```

```
⟨+xpt & dc⟩ \DeclarePreloadSizes{T1}{cmr}{m}{n}{5,7,10}
⟨+xipt & cm⟩ \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,10.95}
⟨+xipt & dc⟩ \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,10.95}
⟨+xiipt & cm⟩ \DeclarePreloadSizes{OT1}{cmr}{m}{n}{6,8,12}
⟨+xiipt & dc⟩ \DeclarePreloadSizes{T1}{cmr}{m}{n}{6,8,12}
%%
%% Computer Modern Sans:
%%----------------------
⟨+ori⟩ \DeclarePreloadSizes{OT1}{cmss}{m}{n}{10,10.95,12}
%%
%% Computer Modern Typewriter:
%%---------------------------
⟨+ori⟩ \DeclarePreloadSizes{OT1}{cmtt}{m}{n}{9,10,10.95,12}
%%
%% Computer Modern Math:
%%----------------------
⟨*ori⟩
\DeclarePreloadSizes{OML}{cmm}{m}{it}
          {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}
          {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
⟨/ori⟩
```

The math fonts are the same for both DC and CM fonts. So far there isn't an agreed on standard.

```
⟨*xpt⟩
\DeclarePreloadSizes{OML}{cmm}{m}{it}{5,7,10}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{5,7,10}
⟨/xpt⟩
⟨*xipt⟩
\DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,10.95}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,10.95}
⟨/xipt⟩
⟨*xiipt⟩
\DeclarePreloadSizes{OML}{cmm}{m}{it}{6,8,12}
\DeclarePreloadSizes{OMS}{cmsy}{m}{n}{6,8,12}
⟨/xiipt⟩
%%
%% LaTeX symbol fonts:
%%--------------------
⟨*ori⟩
\DeclarePreloadSizes{U}{lasy}{m}{n}
          {5,6,7,8,9,10,10.95,12,14.4,17.28,20.74}
⟨/ori⟩
⟨/preload⟩
```

Command	*Corresponds to*	*Action*
`\textrm{..}`	`\rmfamily`	Typeset argument in roman family
`\textsf{..}`	`\sffamily`	Typeset argument in sans serif family
`\texttt{..}`	`\ttfamily`	Typeset argument in `typewriter` family
`\textmd{..}`	`\mdseries`	Typeset argument in medium series
`\textbf{..}`	`\bfseries`	Typeset argument in **bold** series
`\textup{..}`	`\upshape`	Typeset argument in normal shape
`\textit{..}`	`\itshape`	Typeset argument in *italic* shape
`\textsl{..}`	`\slshape`	Typeset argument in *slanted* shape
`\textsc{..}`	`\scshape`	Typeset argument in SMALL CAPS shape
`\emph{..}`	`\em`	Typeset argument *emphasized*

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.

File v

ltfntcmd.dtx

Abstract

The commands defined in this file `ltfntcmd` are part of the kernel code for LaTeX 2ε/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.

49 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 TeX system and for several reasons it is better to avoid them on the user level whenever possible. In LaTeX3 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`.[7] The default definition for this list is produced by the following.

```
\newcommand \nocorrlist {,.}
```

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 LaTeX 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 LaTeX'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 `\/`.

---

[7] 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.

50 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 prceding 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.

```
⟨*2ekernel⟩
\def \DeclareTextFontCommand #1#2{%
  \DeclareRobustCommand#1[1]{%
    \ifmmode
      \nfss@text{#2##1}%
    \else
      \hmode@bgroup
       \text@command{##1}%
       #2\check@icl ##1\check@icr
       \expandafter
      \egroup
    \fi
                        }%
}
```

`\textrm` `\textsf` `\texttt` `\textnormal`

Now we define the `\text`⟨*family*⟩ commands in terms of the above; `\texttt` does not look very nice!

```
\DeclareTextFontCommand{\textrm}{\rmfamily}
\DeclareTextFontCommand{\textsf}{\sffamily}
\DeclareTextFontCommand{\texttt}{\ttfamily}
\DeclareTextFontCommand{\textnormal}{\normalfont}
```

`\textbf` `\textmd`

For the series attribute:

```
\DeclareTextFontCommand{\textbf}{\bfseries}
\DeclareTextFontCommand{\textmd}{\mdseries}
```

`\textit` `\textsl` `\textsc` `\textup`

And for the shapes:

```
\DeclareTextFontCommand{\textit}{\itshape}
\DeclareTextFontCommand{\textsl}{\slshape}
\DeclareTextFontCommand{\textsc}{\scshape}
\DeclareTextFontCommand{\textup}{\upshape}
```

`\emph` Finally we have the `\em` font change declaration of LaTeX. The corresponding definition with argument is

```
\DeclareTextFontCommand{\emph}{\em}
```

`\nocorr` This is just a label, so it does nothing; it should also be unexpandable.

```
\let \nocorr \relax
```

`\check@icl`
`\check@icr` We define these defaults in case some error causes them to be expanded at the wrong time.

```
\let \check@icl \@empty
\let \check@icr \@empty
```

`\text@command`
`\check@nocorr@` 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!

```
\def \text@command #1{%
  \def \reserved@a {#1}%
  \ifx \reserved@a \@empty
    \let \check@icl \@empty
    \let \check@icr \@empty
  \else
```

`\space` is a reserved word in LaTeX or actually already in plain TeX. If somebody really redefines it so many things will break that I don't see any reason to make this routine here slower than necessary.

```
%    \def \reserved@b { }%
%    \ifx \reserved@a \reserved@b
    \ifx \reserved@a \space
      \let \check@icl \@empty
      \let \check@icr \@empty
    \else
      \check@nocorr@ #1\nocorr\@nil
    \fi
  \fi
}
\def \check@nocorr@ #1#2\nocorr#3\@nil {%
```

The two checks are initialised here to their values in the normal case.

```
  \let \check@icl \maybe@ic
  \def \check@icr {\ifvmode \else \aftergroup \maybe@ic \fi}%
  \def \reserved@a {\nocorr}%
  \def \reserved@b {#1}%
  \def \reserved@c {#3}%
  \ifx \reserved@a \reserved@b
    \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.

```
      \let \check@icl \@empty
    \else
```

Otherwise there is a `\nocorr` both at the start and elsewhere, so no italic corrections should be added.

```
      \let \check@icl \@empty
      \let \check@icr \@empty
    \fi
  \else
    \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.

```
    \else
```

In this case there is no `\nocorr` at the start but there is one elsewhere, so no `\aftergroup` is needed.

```
      \let \check@icr \@empty
    \fi
  \fi
}
```

`\ifmaybe@ic` Switch used soley within `\maybe@ic` not interferring with other switches.

```
\newif\ifmaybe@ic
```

`\maybe@ic`
`\maybe@ic@` These macros implement the italic correction.

```
\def \maybe@ic {\futurelet\@let@token\maybe@ic@}
\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?

```
  \ifdim  \fontdimen\@ne\font>\z@
  \else
    \maybe@ictrue
```

It would be possible, but probably not worthwhile, to continue the forward scan beyond any closing braces.

```
    \expandafter\@tfor\expandafter\reserved@a\expandafter:\expandafter=%
        \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.

```
    \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`.

```
    \ifmaybe@ic \sw@slant \fi
  \fi
}
```

`\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.

```
\def \t@st@ic {%
  \expandafter\let\expandafter\reserved@b\expandafter=\reserved@a\relax
  \ifx\reserved@b\@let@token
```

If they are the same we record the fact and jump out of the loop.

```
    \maybe@icfalse
    \@break@tfor
  \fi
}
```

`\sw@slant` `\fix@penalty` The definition of the mysterious `\sw@slant` command is as follows.

```
\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.

```
  \ifdim \lastskip=\z@
    \fix@penalty
  \else
    \skip@ \lastskip
    \unskip
    \fix@penalty
    \hskip \skip@
  \fi
}
```

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 LaTeX code?

```
\def \fix@penalty {%
  \ifnum \lastpenalty=\z@
    \@@italiccorr
  \else
    \count@ \lastpenalty
    \unpenalty
    \@@italiccorr
    \penalty \count@
  \fi
}
```

`\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.

```
\def \nocorrlist {,.}
```

`\nfss@text` This command will by default behave like a LaTeX `\mbox` but may be redefined by packages such as `amstext.sty` to be a bit cleverer.

```
\ifx \nfss@text\@undefined
  \def \nfss@text {\leavevmode\hbox}
\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}
```

```
\def \DeclareOldFontCommand #1#2#3{%
  \DeclareRobustCommand #1{\@fontswitch {#2}{#3}}%
}
```

`\@fontswitch`
`\@@math@egroup`
`\@@math@egroup`

These two commands actually do the necessary tests and declarative font- or alphabet-changing.

```
\def \@fontswitch #1#2{%
  \ifmmode
     \let \math@bgroup \relax
     \def \math@egroup {\let \math@bgroup \@@math@bgroup
                        \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).

```
     #2\relax
  \else
     #1%
  \fi
}
\let \@@math@bgroup \math@bgroup
\let \@@math@egroup \math@egroup
```

These commands are available only in the preamble.

```
\@onlypreamble \DeclareTextFontCommand
\@onlypreamble \DeclareOldFontCommand
```

51 Initialization

`\normalsize`

This is defined to produce an error.

```
\def\normalsize{%
  \@latex@error {The font size command \protect\normalsize\space
               is not defined:\MessageBreak
               there is probably something wrong with
               the class file}\@eha
}
⟨/2ekernel⟩
```

File w

ltpageno.dtx

52 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.

```
⟨*2ekernel⟩
\message{page nos.,}

\countdef\c@page=0 \c@page=1
\def\cl@page{}
\def\pagenumbering#1{%
  \global\c@page \@ne \gdef\thepage{\csname @#1\endcsname
   \c@page}}

⟨/2ekernel⟩
```

File x

ltxref.dtx

53 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.

53.1 Cross Referencing

```
⟨*2ekernel⟩
\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` `\@refundefined` This does not save on name-space (since `\G@refundefinedfalse` was never needed) but it does make the implmentation 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* correspnd 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 TEX 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.

```
% \newif\ifG@refundefined
% \def\G@refundefinedtrue{\global\let\ifG@refundefined\iftrue}
% \def\G@refundefinedfalse{\global\let\ifG@refundefined\iffalse}
\def\G@refundefinedtrue{%
  \gdef\@refundefined{%
    \@latex@warning@no@line{There were undefined references}}}
\let\@refundefined\relax
```

`\ref` `\pageref` `\@setref` Referencing a `\label`. RmS 91/10/25: added a few extra `\reset@font`, as suggested by Bernd Raichle

RmS 92/08/14: made `\ref` and `\pageref` robust

RmS 93/09/08: Added setting of refundefined switch.

```
\def\@setref#1#2#3{%
  \ifx#1\relax
   \protect\G@refundefinedtrue
   \nfss@text{\reset@font\bfseries ??}%
   \@latex@warning{Reference ‘#3’ on page \thepage \space
             undefined}%
  \else
   \expandafter#2#1\null
  \fi}
\def\ref#1{\expandafter\@setref\csname r@#1\endcsname\@firstoftwo{#1}}
\def\pageref#1{\expandafter\@setref\csname r@#1\endcsname
                                    \@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 it's 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.

```
\def\@newl@bel#1#2#3{{%
  \@ifundefined{#1@#2}%
    \relax
    {\gdef \@multiplelabels {%
      \@latex@warning@no@line{There were multiply-defined labels}}%
     \@latex@warning@no@line{Label ‘#2’ multiply defined}}%
  \global\@namedef{#1@#2}{#3}}}
```

```
\def\newlabel{\@newl@bel r}
```

```
\@onlypreamble\@newl@bel
```

`\if@multiplelabels` `\@multiplelabels` This is redefined to produce a warning if at least one label is defined more than once. It is executed by the `\enddocument` command.

```
\let \@multiplelabels \relax
```

`\label` `\refstepcounter` The commands `\label` and `\refstepcounter` have been changed to allow `\protect`'ed commands to work properly. For example,

```
\def\thechapter{\protect\foo{\arabic{chapter}.\roman{section}}}
```

will cause a `\label{bar}` command to define `\ref{bar}` to expand to something like `\foo{4.d}`. Change made 20 Jul 88.

```
\def\label#1{\@bsphack
  \protected@write\@auxout{}%
         {\string\newlabel{#1}{{\@currentlabel}{\thepage}}}%
  \@esphack}
```

```
\def\refstepcounter#1{\stepcounter{#1}%
    \protected@edef\@currentlabel
       {\csname p@#1\endcsname\csname the#1\endcsname}%
}
```

`\@currentlabel` For `\label` commands that come before any environment

```
\def\@currentlabel{}
```

⟨/2ekernel⟩

53.2 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 of 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 exisiting 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 LaTeX 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

54 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`.

```
⟨*2ekernel⟩
\message{environments,}
```

54.1 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 TEX82.

`\enddocument` is the user’s command for ending the manuscript file.

`\stop` is a panic button — to end TEX 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
               \bibcite{LABEL}{VAL} == null
                    BEGIN
                      \reserved@a == VAL
                      if def(\reserved@a) = def(\g@LABEL)
                        else @tempswa := true          fi
                    END
               @tempswa := false
               make @ a letter
               \input \jobname.AUX
               if @tempswa = true
                 then LaTeX Warning: 'Label may have changed.
                                      Rerun to get cross-references right.'
       fi      fi      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.

```
\def\@currenvir{document}
```

`\if@ignore`

`\@ignoretrue`

`\@ignorefalse`

```
\def\@ignorefalse{\global\let\if@ignore\iffalse}
\def\@ignoretrue {\global\let\if@ignore\iftrue}
\@ignorefalse
```

`\ignorespacesafterend`

```
\let\ignorespacesafterend\@ignoretrue
```

`\enddocument`

```
\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…

```
   \let\AtEndDocument\@firstofone
   \@enddocumenthook
   \@checkend{document}%
   \clearpage
   \begingroup
     \if@filesw
       \immediate\closeout\@mainaux
       \let\@setckpt\@gobbletwo
       \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`.

```
       \@tempswafalse
       \makeatletter \@@input\jobname.aux
     \fi

     \@dofilelist
```

First we check for font size substitution bigger than `\fontsubfuzz`. The `\relax` is necessary because this is a macro not a register.

```
     \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.

```
       \@font@warning{Size substitutions with differences\MessageBreak
                  up to \font@submax\space have occurred.\@gobbletwo}%
     \fi
```

The macro `\@defaultsubs` is initially `\relax` but gets redefined to produce a warning if there have been some default font substitutions.

```
    \@defaultsubs
```

The macro `\@refundefined` is initially `\relax` but gets redefined to produce a warning if there are undefined refs.

```
    \@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.

```
    \if@filesw
      \ifx \@multiplelabels \relax
        \if@tempswa
          \@latex@warning@no@line{Label(s) may have changed.
              Rerun to get cross-references right}%
        \fi
      \else
        \@multiplelabels
      \fi
    \fi
  \endgroup
  \deadcycles\z@\@@end}
```

`\@testdef`

```
\def\@testdef #1#2#3{%
  \def\reserved@a{#3}\expandafter \ifx \csname #1@#2\endcsname
 \reserved@a  \else \@tempswatrue \fi}
```

`\@writefile`

```
\long\def\@writefile#1#2{%
  \@ifundefined{tf@#1}\relax
    {\@temptokena{#2}%
     \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
    }%
}
```

`\stop`

```
\def\stop{\clearpage\deadcycles\z@\let\par\@@par\@@end}
```

```
\everypar{\@nodocument} %% To get an error if text appears before the
\nullfont               %% \begin{document}
```

`\begin`, `\end`, and `\@checkend` changed so `\end`{document} will catch an unmatched `\begin`. Changed 24 May 89 as suggested by Frank Mittelbach and Rainer Sch\"opf.

```
 \begin{NAME} ==
  BEGIN
    IF \NAME undefined  THEN  \reserved@a == BEGIN report error
END
                           ELSE  \reserved@a ==
                                      (\@currenvir :=L NAME) \NAME
    FI
    @ignore :=G F       %% Added 30 Nov 88
    \begingroup
    \@endpe := F
    \@currenvir :=L NAME
    \NAME
  END

 \end{NAME} ==
  BEGIN
   \endNAME
   \@checkend{NAME}
   \endgroup
   IF @endpe = T                 %% @endpe set True by \@endparenv
     THEN \@doendpe               %% \@doendpe redefines \par and
\everypar

                                %% to suppress paragraph indentation in
   FI                           %% immediately following text
   IF @ignore = T
     THEN @ignore :=G F
          \ignorespaces
   FI
  END

 \@checkend{NAME} ==
  BEGIN
   IF \@currenvir = NAME
     ELSE \@badend{NAME}
   FI
  END
```

`\begin`

```
\def\begin#1{%
  \@ifundefined{#1}%
    {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
    {\def\reserved@a{\def\@currenvir{#1}%
     \edef\@currenvline{\on@line}%
     \csname #1\endcsname}}%
  \@ignorefalse
  \begingroup\@endpefalse\reserved@a}
```

`\end`

```
\def\end#1{%
  \csname end#1\endcsname\@checkend{#1}%
  \expandafter\endgroup\if@endpe\@doendpe\fi
  \if@ignore\@ignorefalse\ignorespaces\fi}
```

`\@checkend`

```
\def\@checkend#1{\def\reserved@a{#1}\ifx
      \reserved@a\@currenvir \else\@badend{#1}\fi}
```

`\@currenvline` We do need a default value for `\@currenvline` on top-level since the document environment cancels the brace group. This means that a missmatch with `\begin {document}` will not produce a line number. Thus the outer default must be `\@empty` or we will end up with two spaces.

```
\let\@currenvline\@empty
```

54.2 Center, Flushright, Flushleft

```
\message{center,}
```

```
 \center, \flushright and \flushleft set
   \rightskip = 0pt or \@flushglue (as appropriate)
```

```
\leftskip  = 0pt or \@flushglue (as appropriate)
\parindent = 0pt
\parfillskip   = 0pt. (except \flushleft)
\\         == \par \vskip -\parskip
\\[LENGTH] == \\ \vskip LENGTH
\\*        == \par \penalty 10000 \vskip -\parskip
\\*[LEN]   == \\* \vskip LENGTH
```

They invoke the trivlist environment to handle vertical spacing before and after them.

`\centering`, `\raggedright` and `\raggedleft` are the declaration analogs of the above.

`\raggedright` has a more universal effect, however. It sets `\@rightskip` := flushglue. Every environment, like the list environments,
that set `\rightskip` to its 'normal' value set it to `\@rightskip`

`\@centercr`

```
\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
       \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
```

`\@xcentercr`

```
\def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
    [\@icentercr\ignorespaces}
```

`\@icentercr`

```
\def\@icentercr[#1]{\vskip #1\ignorespaces}
```

`center` We use `\relax` to prevent `\item` scanning too far.

```
\def\center{\trivlist \centering\item\relax}
```

```
\def\endcenter{\endtrivlist}
```

`\centering`

```
\def\centering{%
  \let\\\@centercr
  \rightskip\@flushglue\leftskip\@flushglue
  \parindent\z@\parfillskip\z@skip}
```

`\@rightskip`

```
\newskip\@rightskip \@rightskip \z@skip
```

`flushleft` We use `\relax` to prevent `\item` scanning too far.

```
\def\flushleft{\trivlist \raggedright\item\relax}
```

```
\def\endflushleft{\endtrivlist}
```

`\raggedright`

```
\def\raggedright{%
  \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
  \leftskip\z@skip
  \parindent\z@}
```

`flushright` We use `\relax` to prevent `\item` scanning too far.

```
\def\flushright{\trivlist \raggedleft\item\relax}
```

```
\def\endflushright{\endtrivlist}
```

`\raggedleft`

```
\def\raggedleft{%
  \let\\\@centercr
  \rightskip\z@skip\leftskip\@flushglue
  \parindent\z@\parfillskip\z@skip}
```

```
\message{verbatim,}
```

54.3 Verbatim

The verbatim environment uses the fixed-width `\ttfamily` font, turns blanks into spaces, starts a new line for each carrige return (or sequence of consecutive carriage returns), and interprets *every* character literally. I.e., all special characters `\`, `{`, `$`, etc. are `\catcode`'d to 'other'.

The command `\verb` produces in-line verbatim text, where the argument is delimited by any pair of characters. E.g., `\verb #...#` takes '`...`' as its argument, and sets it verbatim in `\ttfamily` font.

The *-variants of these commands are the same, except that spaces print as the TeXbook's space character instead of as blank spaces.

`\@vobeyspaces`

```
{\catcode`\ =\active%
\gdef\@vobeyspaces{\catcode`\ \active\let \@xobeysp}}
```

`\@xobeysp`

`\@xverbatim`

`\@sxverbatim`

```
\begingroup \catcode `|=0 \catcode `[= 1
\catcode`]=2 \catcode `\{=12 \catcode `\}=12
\catcode`\\=12 |gdef|@xverbatim#1\end{verbatim}[#1|end[verbatim]]
|gdef|@sxverbatim#1\end{verbatim*}[#1|end[verbatim*]]
|endgroup
```

`\@verbatim` Real start of verbatim environment We use `\relax` to prevent `\item` scanning too far.

```
\def\@verbatim{\trivlist \item\relax
  \if@minipage\else\vskip\parskip\fi
  \leftskip\@totalleftmargin\rightskip\z@skip
  \parindent\z@\parfillskip\@flushglue\parskip\z@skip
```

Added `\@@par` to clear possible `\parshape` definition from a surrounding list (the verbatim guru says).

```
  \@@par
  \@tempswafalse
  \def\par{%
    \if@tempswa
```

A `\leavevmode` added: needed if, for example, a blank verbatim line is the first thing in a list item (wow!).

```
      \leavevmode \null \@@par\penalty\interlinepenalty
    \else
      \@tempswatrue
      \ifhmode\@@par\penalty\interlinepenalty\fi
    \fi}%
```

To allow customization we hide the font used in a separate macro.

```
  \let\do\@makeother \dospecials
  \obeylines \verbatim@font \@noligs
  \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`!

```
  \everypar \expandafter{\the\everypar \unpenalty}%
}
```

`\verbatim`
`\endverbatim`

(RmS 93/09/19) Protected against ‘missing item’ error message triggered by empty verbatim environment.

```
\def\verbatim{\@verbatim \frenchspacing\@vobeyspaces \@xverbatim}
\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.

```
\def\verbatim@font{\normalfont\ttfamily}
```

`verbatim*`

```
\@namedef{verbatim*}{\@verbatim\@sxverbatim}
\expandafter\let\csname endverbatim*\endcsname =\endverbatim
```

`\@makeother`

```
\def\@makeother#1{\catcode‘#112\relax}
```

`\verb@balance@group`

```
\let\verb@balance@group\@empty
```

`\verb@egroup`

```
\def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
```

`\verb@eol@error`

```
\begingroup
  \obeylines%
  \gdef\verb@eol@error{\obeylines%
    \def^^M{\verb@egroup\@latex@error{%
             \noexpand\verb ended by end of line}\@ehc}}%
\endgroup
```

`\verb`

Typesetting a small piece verbatim.

```
\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
  \bgroup
    \verb@eol@error \let\do\@makeother \dospecials
    \verbatim@font\@noligs
    \@ifstar\@sverb\@verb}
```

`\@sverb`

Definitions of `\@sverb` and `\@verb` changed so `\verb+ foo+` does not lose leading blanks when it comes at the beginning of a line. Change made 24 May 89. Suggested by Frank Mittelbach and Rainer Schöpf.

```
\def\@sverb#1{%
  \catcode‘#1\active
  \lccode‘\~‘#1%
  \gdef\verb@balance@group{\verb@egroup
     \@latex@error{\noexpand\verb illegal in command argument}\@ehc}%
  \aftergroup\verb@balance@group
  \lowercase{\let~\verb@egroup}}%
```

`\@verb`

```
\def\@verb{\@vobeyspaces \frenchspacing \@sverb}
```

`\verbatim@nolig@list`

```
\def\verbatim@nolig@list{\do\‘\do\<\do\>\do\,\do\’\do\-}
```

\do@noligs

```
\def\do@noligs#1{%
  \catcode`#1\active
  \begingroup
     \lccode`\~`#1\relax
     \lowercase{\endgroup\def~{\leavevmode\kern\z@\char`#1}}}
```

\@noligs To stay compatible with packages that use `\@noligs` we keep it.

```
\def\@noligs{\let\do\do@noligs \verbatim@nolig@list}
```

```
⟨/2ekernel⟩
```

File z

ltmath.dtx

55 Math setup

This file contains a lot of the original plain TEX code, as well as the LATEX environments for math. It still needs sorting out.

```
⟨*2ekernel⟩
\message{math definitions,}
```

55.1 Math commands based on plain TEX

55.1.1 The log-like functions

`\log` The standard operators:

```
\def\log{\mathop{\operator@font log}\nolimits}
\def\lg{\mathop{\operator@font lg}\nolimits}
\def\ln{\mathop{\operator@font ln}\nolimits}
\def\lim{\mathop{\operator@font lim}}
\def\limsup{\mathop{\operator@font lim\,sup}}
\def\liminf{\mathop{\operator@font lim\,inf}}
\def\sin{\mathop{\operator@font sin}\nolimits}
\def\arcsin{\mathop{\operator@font arcsin}\nolimits}
\def\sinh{\mathop{\operator@font sinh}\nolimits}
\def\cos{\mathop{\operator@font cos}\nolimits}
\def\arccos{\mathop{\operator@font arccos}\nolimits}
\def\cosh{\mathop{\operator@font cosh}\nolimits}
\def\tan{\mathop{\operator@font tan}\nolimits}
\def\arctan{\mathop{\operator@font arctan}\nolimits}
\def\tanh{\mathop{\operator@font tanh}\nolimits}
\def\cot{\mathop{\operator@font cot}\nolimits}
\def\coth{\mathop{\operator@font coth}\nolimits}
\def\sec{\mathop{\operator@font sec}\nolimits}
\def\csc{\mathop{\operator@font csc}\nolimits}
\def\max{\mathop{\operator@font max}}
\def\min{\mathop{\operator@font min}}
\def\sup{\mathop{\operator@font sup}}
\def\inf{\mathop{\operator@font inf}}
\def\arg{\mathop{\operator@font arg}\nolimits}
\def\ker{\mathop{\operator@font ker}\nolimits}
\def\dim{\mathop{\operator@font dim}\nolimits}
\def\hom{\mathop{\operator@font hom}\nolimits}
\def\det{\mathop{\operator@font det}}
\def\exp{\mathop{\operator@font exp}\nolimits}
\def\Pr{\mathop{\operator@font Pr}}
\def\gcd{\mathop{\operator@font gcd}}
\def\deg{\mathop{\operator@font deg}\nolimits}
```

`\bmod` And some operators have to be done by hand:

```
\def\bmod{%
  \nonscript\mskip-\medmuskip\mkern5mu%
  \mathbin{\operator@font mod}\penalty900\mkern5mu%
  \nonscript\mskip-\medmuskip}
```

`\pmod`

```
\def\pmod#1{%
  \allowbreak\mkern18mu({\operator@font mod}\,\,#1)}
```

55.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```
\def\bigl{\mathopen\big}
\def\bigm{\mathrel\big}
\def\bigr{\mathclose\big}
\def\Bigl{\mathopen\Big}
\def\Bigm{\mathrel\Big}
\def\Bigr{\mathclose\Big}
\def\biggl{\mathopen\bigg}
\def\biggm{\mathrel\bigg}
\def\biggr{\mathclose\bigg}
\def\Biggl{\mathopen\Bigg}
\def\Biggm{\mathrel\Bigg}
\def\Biggr{\mathclose\Bigg}
```

55.1.3 The UNSORTED Rest

The other math commands are lifted from plain TEX.

`\jot`

```
\newdimen\jot
\jot=3pt
```

`\interdisplaylinepenalty`

```
\newcount\interdisplaylinepenalty
\interdisplaylinepenalty=100
```

`\choose`

```
\def\choose{\atopwithdelims()}
```

`\brack`

```
\def\brack{\atopwithdelims[]}
```

`\brace`

```
\def\brace{\atopwithdelims\{\}}
```

`\mathpalette`

```
\def\mathpalette#1#2{%
  \mathchoice
    {#1\displaystyle{#2}}%
    {#1\textstyle{#2}}%
    {#1\scriptstyle{#2}}%
    {#1\scriptscriptstyle{#2}}}
```

`\root`
`\rootbox`
`\r@@t`

```
\newbox\rootbox
```

```
\def\root#1\of{%
  \setbox\rootbox\hbox{$\m@th\scriptscriptstyle{#1}$}%
  \mathpalette\r@@t}
```

```
\def\r@@t#1#2{%
  \setbox\z@\hbox{$\m@th#1\sqrtsign{#2}$}%
  \dimen@\ht\z@ \advance\dimen@-\dp\z@
  \mkern5mu\raise.6\dimen@\copy\rootbox
  \mkern-10mu\box\z@}
```

`\phantom`
`\hphantom`
`\vphantom`

```
\newif\ifv@
\newif\ifh@
```

```
\def\vphantom{\v@true\h@false\ph@nt}
```

```
\def\hphantom{\v@false\h@true\ph@nt}
```

```
\def\phantom{\v@true\h@true\ph@nt}
```

```
\def\ph@nt{%
  \ifmmode
    \expandafter\mathpalette\expandafter\mathph@nt
  \else
    \expandafter\makeph@nt
  \fi}
```

```
\def\makeph@nt#1{%
  \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finph@nt}
```

```
\def\mathph@nt#1#2{%
  \setbox\z@\hbox{$\m@th#1{#2}$}\finph@nt}
```

```
\def\finph@nt{%
  \setbox\tw@\null
  \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
  \ifh@ \wd\tw@\wd\z@\fi \box\tw@}
```

`\mathstrut`

```
\def\mathstrut{\vphantom(}
```

`\smash`

```
\def\smash{%
  \relax % \relax, in case this comes first in \halign
  \ifmmode
    \expandafter\mathpalette\expandafter\mathsm@sh
  \else
    \expandafter\makesm@sh
  \fi}
```

```
\def\makesm@sh#1{%
  \setbox\z@\hbox{\color@begingroup#1\color@endgroup}\finsm@sh}
\def\mathsm@sh#1#2{%
  \setbox\z@\hbox{$\m@th#1{#2}$}\finsm@sh}
\def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}
```

`\buildrel`

```
\def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}
```

`\cases`

```
\def\cases#1{\left\{\,\vcenter{\normalbaselines\m@th
    \ialign{$##\hfil$&\quad{##}\hfil\crcr#1\crcr}}\right.}
```

`\matrix`

```
\def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
    \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crcr
      \mathstrut\crcr\noalign{\kern-\baselineskip}
      #1\crcr\mathstrut\crcr\noalign{\kern-\baselineskip}}}\,}
```

`\pmatrix`

```
\def\pmatrix#1{\left(\matrix{#1}\right)}
```

`\bordermatrix`

```
\def\bordermatrix#1{\begingroup \m@th
  \@tempdima 8.75\p@
  \setbox\z@\vbox{%
    \def\cr{\crcr\noalign{\kern2\p@\global\let\cr\endline}}%
    \ialign{$##$\hfil\kern2\p@\kern\@tempdima&\thinspace\hfil$##$\hfil
      &&\quad\hfil$##$\hfil\crcr
      \omit\strut\hfil\crcr\noalign{\kern-\baselineskip}%
      #1\crcr\omit\strut\cr}}%
  \setbox\tw@\vbox{\unvcopy\z@\global\setbox\@ne\lastbox}%
  \setbox\tw@\hbox{\unhbox\@ne\unskip\global\setbox\@ne\lastbox}%
  \setbox\tw@\hbox{$\kern\wd\@ne\kern-\@tempdima\left(\kern-\wd\@ne
    \global\setbox\@ne\vbox{\box\@ne\kern2\p@}%
    \vcenter{\kern-\ht\@ne\unvbox\z@\kern-\baselineskip}\,\right)$}%
  \null\;\vbox{\kern\ht\@ne\box\tw@}\endgroup}
```

`\openup`

```
\def\openup{\afterassignment\@penup\dimen@}
\def\@penup{\advance\lineskip\dimen@
  \advance\baselineskip\dimen@
  \advance\lineskiplimit\dimen@}
```

`\displaylines`

```
\newif\ifdt@p
\def\displ@y{\global\dt@ptrue\openup\jot\m@th
  \everycr{\noalign{\ifdt@p \global\dt@pfalse \ifdim\prevdepth>-1000\p@
      \vskip-\lineskiplimit \vskip\normallineskiplimit \fi
      \else \penalty\interdisplaylinepenalty \fi}}}
\def\@lign{\tabskip\z@skip\everycr{}} % restore inside \displ@y
\def\displaylines#1{\displ@y \tabskip\z@skip
  \halign{\hb@xt@\displaywidth{$\@lign\hfil\displaystyle##\hfil$}\crcr
    #1\crcr}}
```

`\sp`
`\sb`

```
\let\sp=^
\let\sb=_
```

`\>`
`\;`
`\!`

```
%\def\,{\mskip\thinmuskip}     % already defined in ltspace
\def\>{\mskip\medmuskip}
\def\;{\mskip\thickmuskip}
\def\!{\mskip-\thinmuskip}
```

`\*`

```
\def\*{\discretionary{\thinspace\the\textfont2\char2}{}{}}
```

`\:` Nickname for the medium space since `\>` is not available inside `tabbing`.

```
\let\:=\>
```

`\active@math@prime` This is the definition of the active math prime.

```
\def\active@math@prime{^\bgroup\prim@s}
```

`\prime@s`

```
{\catcode‘\’=\active \global\let’\active@math@prime}
\def\prim@s{%
  \prime\futurelet\@let@token\pr@m@s}
\def\pr@m@s{%
  \ifx’\@let@token
    \expandafter\pr@@@s
  \else
    \ifx^\@let@token
      \expandafter\expandafter\expandafter\pr@@@t
    \else
      \egroup
    \fi
  \fi}
```

```
\def\pr@@@s#1{\prim@s}
```

```
\def\pr@@@t#1#2{#2\egroup}
```

```
{\catcode‘\_=\active \gdef_{\_}} % _ in math is
                                 % either subscript or \_
```

55.2 Math Environments

`\(` `\)` Produces `$...$` with checks that `\(` isn't used in math mode, and that `\)` is only used in math mode begun with `\(`.

```
\def\({\relax\ifmmode\@badmath\else$\fi}
\def\){\relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}
```

`\[` `\]` Produces `$$...$$` with checks that `\[` isn't used in math mode, and that `\]` is only used in math mode begun with `\]`.

```
\def\[{%
   \relax\ifmmode
      \@badmath
   \else
      \ifvmode
         \nointerlineskip
         \makebox[.6\linewidth]{}%
      \fi
      $$%%$$ BRACE MATCH HACK
   \fi
}
```

```
\def\]{%
   \relax\ifmmode
      \ifinner
         \@badmath
      \else
         $$%%$$ BRACE MATCH HACK
      \fi
   \else
      \@badmath
   \fi
   \ignorespaces
}
```

`math` `displaymath` Disguises for `\(...\)` and `\[...\]`.

```
\let\math=\(
\let\endmath=\)
```

```
\def\displaymath{\[}
\def\enddisplaymath{\]\@ignoretrue}
```

`equation` `\c@equation` Numbered equations, using the counter `\c@equation`. *Note*: The document style must define `\theequation` etc., and do the appropriate `\@addtoreset`. It should also redefine `\@eqnnum` if another format for the equation number is desired other than the standard (...), or to move the equation numbers to the flushleft. (See comment on the `\def` of `\@eqnnum`.)

```
\@definecounter{equation}
\def\equation{$$\refstepcounter{equation}}
\def\endequation{\eqno \hbox{\@eqnnum}$$\@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.

```
\def\@eqnnum{{\normalfont \normalcolor (\theequation)}}
```

`\stackrel` A disguise for plain TeX's buildrel.

```
\def\stackrel#1#2{\mathrel{\mathop{#2}\limits^{#1}}}
```

`\frac` A disguise for plain TeX's `\over`.

```
\def\frac#1#2{{\begingroup#1\endgroup\over#2}}
```

`\sqrt` `\@sqrt` Add an optional argument to plain's `\sqrt` to give the nth root of an expression $\sqrt[n]{e}$.

```
\DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}
\def\@sqrt[#1]{\root #1\of}
```

`eqnarray` `\@eqcnt` `\@eqpen` `\if@eqnsw` `\@eqnsel` Here's the eqnarray environment: Default is for left-hand side of equations to be flushright. To make them flushleft, `\let\@eqnsel = \hfil`.

```
\newcount\@eqcnt
\newcount\@eqpen
\newif\if@eqnsw\@eqnswtrue
\newskip\@centering
\@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.

```
\def\eqnarray{%
   \stepcounter{equation}%
   \def\@currentlabel{\p@equation\theequation}%
   \global\@eqnswtrue
   \m@th
   \global\@eqcnt\z@
   \tabskip\@centering
   \let\\\@eqncr
   $$\everycr{}\halign to\displaywidth\bgroup
       \hskip\@centering$\displaystyle\tabskip\z@skip{##}$\@eqnsel
      &\global\@eqcnt\@ne\hskip \tw@\arraycolsep \hfil${##}$\hfil
      &\global\@eqcnt\tw@ \hskip \tw@\arraycolsep
         $\displaystyle{##}$\hfil\tabskip\@centering
      &\global\@eqcnt\thr@@ \hb@xt@\z@\bgroup\hss##\egroup
         \tabskip\z@skip
      \cr
}
```

```
\def\endeqnarray{%
      \@@eqncr
      \egroup
      \global\advance\c@equation\m@ne
   $$\@ignoretrue
}
```

```
\let\@eqnsel=\relax
```

`\nonumber` Switches off equation numbering.

```
\def\nonumber{\global\@eqnswfalse}
```

`\@eqncr` `\@xeqncr` `\@yeqncr`

```
\def\@eqncr{%
   {\ifnum0=`}\fi
   \@ifstar{%
      \global\@eqpen\@M\@yeqncr
   }{%
      \global\@eqpen\interdisplaylinepenalty \@yeqncr
   }%
}
```

```
\def\@yeqncr{\@testopt\@xeqncr\z@skip}
```

```
\def\@xeqncr[#1]{%
   \ifnum0=`{\fi}%
   \@@eqncr
   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
}
```

`\@@eqncr`

```
\def\@@eqncr{\let\reserved@a\relax
    \ifcase\@eqcnt \def\reserved@a{& & &}\or \def\reserved@a{& &}%
     \or \def\reserved@a{&}\else
       \let\reserved@a\@empty
       \@latex@error{Too many columns in eqnarray environment}\@ehc\fi
     \reserved@a \if@eqnsw\@eqnnum\stepcounter{equation}\fi
     \global\@eqnswtrue\global\@eqcnt\z@\cr}
```

`eqnarray*` Here's the eqnarray* environment:

`\@seqncr`

```
\let\@seqncr=\@eqncr
```

```
\@namedef{eqnarray*}{\def\@eqncr{\nonumber\@seqncr}\eqnarray}
```

```
\@namedef{endeqnarray*}{\nonumber\endeqnarray}
```

`\lefteqn` `\lefteqn{FORMULA}` typesets `FORMULA` in display math style flushleft in a box of width zero.

```
\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.

```
\DeclareRobustCommand{\ensuremath}{%
  \ifmmode
    \expandafter\@firstofone
  \else
    \expandafter\@ensuredmath
  \fi}
```

`\@ensuredmath` The `\relax` stops `\ensuremath{}` starting display math.

```
\long\def\@ensuredmath#1{$\relax#1$}
```

```
⟨/2ekernel⟩
```

55.3 External options to the standard document classes

55.3.1 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 quation is too wide, the equation number than may overprint the equation itself.

```
⟨*leqno⟩
\renewcommand\@eqnnum{\hb@xt@.01\p@{}%
                      \rlap{\normalfont\normalcolor
                        \hskip -\displaywidth(\theequation)}}
⟨/leqno⟩
```

55.3.2 Flush left equations

To get the displayed math environments to print the contents flush left (with an indentation) we have to redefine all of LaTeX 2ε's displayed math environments.

`\mathindent` The amount of indentation of the equations is stored in a register.

```
⟨*fleqn⟩
\newdimen\mathindent
```

The setting of `\mathindent` has to be deferred until the class file has been processed, because `\leftmargini` is still 0pt wide at the moment `fleqn.clo` is read in.

```
\AtEndOfClass{\mathindent\leftmargini}
```

`\[` Begin display math;

```
\renewcommand\[{\relax
               \ifmmode\@badmath
               \else
                 \begin{trivlist}%
                   \@beginparpenalty\predisplaypenalty
                   \@endparpenalty\postdisplaypenalty
                   \item[]\leavevmode
                   \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
                     \hskip\mathindent\bgroup
               \fi}
```

`\]` end display math;

```
\renewcommand\]{\relax
               \ifmmode
                     \egroup $\hfil% $
                   \egroup
                 \end{trivlist}%
               \else \@badmath
               \fi}
```

`equation` The equation environment

```
\renewenvironment{equation}%
    {\@beginparpenalty\predisplaypenalty
     \@endparpenalty\postdisplaypenalty
     \refstepcounter{equation}%
     \trivlist \item[]\leavevmode
       \hb@xt@\linewidth\bgroup $\m@th% $
         \displaystyle
         \hskip\mathindent}%
        {$\hfil % $
         \displaywidth\linewidth\hbox{\@eqnnum}%
       \egroup
     \endtrivlist}
```

`eqnarray` The eqnarray environment

```
\renewenvironment{eqnarray}{%
    \stepcounter{equation}%
    \def\@currentlabel{\p@equation\theequation}%
    \global\@eqnswtrue\m@th
    \global\@eqcnt\z@
    \tabskip\mathindent
    \let\\=\@eqncr
    \setlength\abovedisplayskip{\topsep}%
    \ifvmode
       \addtolength\abovedisplayskip{\partopsep}%
    \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`.

```
    \addtolength\abovedisplayskip{\parskip}%
    \setlength\belowdisplayskip{\abovedisplayskip}%
    \setlength\belowdisplayshortskip{\abovedisplayskip}%
    \setlength\abovedisplayshortskip{\abovedisplayskip}%
    $$\everycr{}\halign to\linewidth% $$
    \bgroup
      \hskip\@centering
      $\displaystyle\tabskip\z@skip{##}$\@eqnsel&%
      \global\@eqcnt\@ne \hskip \tw@\arraycolsep \hfil${##}$\hfil&%
      \global\@eqcnt\tw@ \hskip \tw@\arraycolsep
        $\displaystyle{##}$\hfil \tabskip\@centering&%
      \global\@eqcnt\thr@@
        \hb@xt@\z@\bgroup\hss##\egroup\tabskip\z@skip\cr}%
      {\@@eqncr
    \egroup
    \global\advance\c@equation\m@ne$$% $$
    \@ignoretrue
    }
⟨/fleqn⟩
```

File A

ltlists.dtx

56 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 flexability 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.

56.1 List and Trivlist

The following variables are used inside a list environment:

`\@totalleftmargin` The distance that the prevailing left margin is indented from the outermost left margin,

`\linewidth` The width of the current line. Must be initialized to `\hsize`.

`\@listdepth` A count for holding current list nesting depth.

`\makelabel` A macro with a single argument, used to generate the label from the argument (given or implied) of the `\item` command. Initialized to `\@mklab` by the `\list` command. This command must produce some stretch—i.e., an `\hfil`.

`\@inlabel` A switch that is false except between the time an `\item` is encountered and the time that TeX actually enters horizontal mode. Should be tested by commands that can be messed up by the list environment's use of `\everypar`.

`\box\@labels` When `@inlabel = true`, it holds the labels to be put out by `\everypar`.

`@noparitem` A switch set by `\list` when `@inlabel = true`. Handles the case of a `\list` being the first thing in an item.

`@noparlist` A switch set true for a list that begins an item. No `\topsep` space is added before or after `\item`'s such a list.

`@newlist` Set true by `\list`, set false by the first text (by `\everypar`).

`@noitemarg` Set true when executing an `\item` with no explicit argument. Used to save space. To save time, make two separate `\@item` commands.

`@nmbrlist` Set true by `\usecounter` command, causes list to be numbered.

`\@listctr` `\def`'ed by `\usecounter` to name of counter.

`\@noskipsec` A switch set true by a sectioning command when it is creating an in-text heading with `\everypar`.

Throughout a list environment, `\hsize` is the width of the current line, measured from the outermost left margin to the outermost right margin. Environments like tabbing should use `\linewidth` instead of `\hsize`.

Here are the parameters of a list that can be set by commands in the `\list`'s COMMANDS argument. These parameters are all TeX skips or dimensions (defined by `\newskip` or `\newdimen`), so the usual TeX or LaTeX 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.

56.2 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.

56.3 Penalties

`\@beginparpenalty:` put at the beginning of a list

`\@endparpenalty:` put at end of list

`\@itempenalty:` put between items.

56.4 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 <= `\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.

56.5 Default Values

Defaults for the list environment are set as follows. First, `\rightmargin`, `\listparindent` and `\itemindent` are set to 0pt. Then, one of the commands `\@listi`, `\@listii`, ... , `\@listvi` is called, depending upon the current level of the list. The `\@list` ... commands should be defined by the document style. A convention that the document style should follow is to set `\leftmargin` to `\leftmargini`,..., `\leftmarginvi` for the appropriate level. Items that aren't changed may be left alone, but everything that could possibly be changed must be reset.

```
 \list{LABEL}{COMMANDS} ==
  BEGIN
    if \@listdepth > 5
      then LaTeX error: 'Too deeply nested'
      else \@listdepth :=G \@listdepth + 1
    fi
    \rightmargin     := 0pt
    \listparindent   := 0pt
    \itemindent      := 0pt
    \eval(@list \romannumeral\the\@listdepth)  %% Set default values:
    \@itemlabel      :=L LABEL
    \makelabel       == \@mklab
    @nmbrlist        :=L false
    COMMANDS

    \@trivlist                 % commands common to \list and
\trivlist

    \parskip          :=L \parsep
    \parindent        :=L \listparindent
    \linewidth        :=L \linewidth - \rightmargin -\leftmargin
    \@totalleftmargin :=L \@totalleftmargin + \leftmargin
    \parshape 1 \@totalleftmargin \linewidth
```

```
     \ignorespaces                     % gobble space up to \item
    END

 \endlist == BEGIN \@listdepth :=G \@listdepth -1
                   \endtrivlist
             END

 \@trivlist ==
  BEGIN
     if @newlist = T then \@noitemerr fi
                     %% This command removed for some forgotten
reason.
     \@topsepadd :=L \topsep
     if @noskipsec then leave vertical mode fi  %% Added 11 Jun 85
     if vertical mode
       then \@topsepadd :=L \@topsepadd + \partopsep
       else \unskip \par             % remove glue from end of last line
     fi
     if @inlabel = true
        then @noparitem :=L true
             @noparlist :=L true
        else @noparlist :=L false
             \@topsep   :=L \@topsepadd
     fi
     \@topsep         :=L \@topsep + \parskip  %% Change 4 Sep 85
     \leftskip        :=L 0pt          % Restore paragraphing
parameters
     \rightskip       :=L \@rightskip
     \parfillskip     :=L 0pt + 1fil

   NOTE: \@setpar called on every \list in case \par has been
   temporarily  munged before the \list command.
     \@setpar{if @newlist = false then {\@@par} fi}
     \@newlist          :=G T
     \@outerparskip     :=L \parskip
 END

 \trivlist  ==
 BEGIN
  \parsep     := \parskip
  @nmbrlist := F
  \@trivlist
  \labelwidth := 0
  \leftmargin := 0
  \itemindent := \parindent
  \@itemlabel :=L "empty"          %% added 93/12/13
  \makelabel{LABEL} == LABEL
 END

 \endtrivlist ==
   BEGIN
     if @inlabel = T then \indent fi
     if horizontal mode then \unskip \par fi
     if @noparlist = true
       else if \lastskip > 0
               then \@tempskipa := \lastskip
                    \vskip - \lastskip
                    \vskip \@tempskipa -\@outerparskip + \parskip
           fi
           \@endparenv
    fi
  END

\@endparenv ==
  BEGIN
   \addpenalty{@endparpenalty}
   \addvspace{\@topsepadd}
   \endgroup    %% ends the \begin command's \begingroup
   \par  ==  BEGIN
               \@restorepar
               \everypar{}
               \par
             END
   \everypar == BEGIN remove \lastbox \everypar{} END
   \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 @noparitem = true
      then @noparitem := false
               % NOTE: then clause  hardly every taken,
               %  so made a macro \@donoparitem
           \box\@labels :=G \hbox{\hskip -\leftmargin
                                   \box\@labels
                                   \hskip \leftmargin }
           if @minipage = false then
              \@tempskipa := \lastskip
              \vskip -\lastskip
              \vskip \@tempskipa + \@outerparskip - \parskip
           fi
      else if @inlabel = true
             then \indent \par   % previous item empty.
           fi
           if hmode then 2 \unskip's
                           % To remove any space at end of prev.
                           % paragraph that could cause a blank line.
                    \par
           fi
           if @newlist = T
              then if @nobreak = T   % Kludge if list follows \section
                     then \addvspace{\@outerparskip - \parskip}
                     else \addpenalty{\@beginparpenalty}
                          \addvspace{\@topsep}
                          \addvspace{-\parskip}   %% added 4 Sep 85
                   fi
              else \addpenalty{\@itempenalty}
```

```
                       \addvspace{\itemsep}
               fi
               @inlabel :=G true
       fi

       \everypar{ @minipage :=G F
                  @newlist :=G F
                  if  @inlabel = true
                     then @inlabel :=G false
                          \hskip -\parindent
                          \box\@labels
                          \penalty 0
                                %% 3 Oct 85  – allow line break here
                          \box\@labels :=G null
                  fi
                  \everypar{} }
       @nobreak :=G false
       if  @noitemarg = true
          then @noitemarg := false
               if @nmbrlist
                  then \refstepcounter{\@listctr}
       fi    fi
       \@tempboxa   :=L \hbox{\makelabel{LAB}}
       \box\@labels :=G \@labels \hskip \itemindent
                          \hskip - (\labelwidth + \labelsep)
                          if \wd \@tempboxa > \labelwidth
                             then \box\@tempboxa
                             else \hbox to \labelwidth
{\unhbox\@tempboxa}
                          fi
                          \hskip\labelsep
       \ignorespaces                          %gobble space up to text
    END

    \makelabel{LABEL} == ERROR   %% default to catch lonely \item


    \usecounter{CTR} == BEGIN  @nmbrlist :=L true
                                 \@listctr == CTR
                                 \setcounter{CTR}{0}
                          END
```

DEFINE `\dimen`'s and `\count`

`\topskip`
`\partopsep`
`\itemsep`
`\parsep`
`\@topsep`
`\@topsepadd`
`\outerparskip`

```
⟨*2ekernel⟩
\newskip\topsep
\newskip\partopsep
\newskip\itemsep
\newskip\parsep
\newskip\@topsep
\newskip\@topsepadd
\newskip\@outerparskip
```

`\leftmargin`
`\rightmargin`
`\listparindent`
`\itemindent`
`\labelwidth`
`\labelsep`
`\@totalleftmargin`

```
\newdimen\leftmargin
\newdimen\rightmargin
\newdimen\listparindent
```

```
\newdimen\itemindent
\newdimen\labelwidth
\newdimen\labelsep
\newdimen\linewidth
\newdimen\@totalleftmargin \@totalleftmargin=\z@
```

\leftmargini
\leftmarginii
\leftmarginiii
\leftmarginiv
\leftmarginv
\leftmarginvi

```
\newdimen\leftmargini
\newdimen\leftmarginii
\newdimen\leftmarginiii
\newdimen\leftmarginiv
\newdimen\leftmarginv
\newdimen\leftmarginvi
```

\@listdepth
\@itempenalty
\@beginparpenalty
\@endparpenalty

```
\newcount\@listdepth \@listdepth=0
\newcount\@itempenalty
\newcount\@beginparpenalty
\newcount\@endparpenalty
```

\@labels

```
\newbox\@labels
```

\if@inlabel
\@inlabelfalse
\@inlabeltrue

```
\newif\if@inlabel \@inlabelfalse
```

\if@newlist
\@newlistfalse
\@newlisttrue

```
\newif\if@newlist   \@newlistfalse
```

\if@noparitem
\@noparitemfalse
\@noparitemtrue

```
\newif\if@noparitem \@noparitemfalse
```

\if@noparlist
\@noparlistfalse
\@noparlisttrue

```
\newif\if@noparlist \@noparlistfalse
```

\if@noitemarg
\@noitemargfalse
\@noitemargtrue

```
\newif\if@noitemarg \@noitemargfalse
```

\if@newlist
\@newlistfalse
\@newlisttrue

```
\newif\if@nmbrlist  \@nmbrlistfalse
```

\list

```
\def\list#1#2{%
  \ifnum \@listdepth >5\relax
    \@toodeep
  \else
    \global\advance\@listdepth\@ne
  \fi
  \rightmargin\z@
  \listparindent\z@
  \itemindent\z@
  \csname @list\romannumeral\the\@listdepth\endcsname
  \def\@itemlabel{#1}%
  \let\makelabel\@mklab
  \@nmbrlistfalse
  #2\relax
  \@trivlist
  \parskip\parsep
  \parindent\listparindent
  \advance\linewidth -\rightmargin
  \advance\linewidth -\leftmargin
  \advance\@totalleftmargin \leftmargin
  \parshape \@ne \@totalleftmargin \linewidth
  \ignorespaces}
```

`\par@deathcycles`

```
\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 TeX. We thus need to trap this here by seting `\par` to count the number of times a paragraph ii is called with no progress being made started.

```
\def\@trivlist{%
  \if@noskipsec \leavevmode \fi
  \@topsepadd \topsep
  \ifvmode
    \advance\@topsepadd \partopsep
  \else
    \unskip \par
  \fi
  \if@inlabel
    \@noparitemtrue
    \@noparlisttrue
  \else
    \if@newlist \@noitemerr \fi
    \@noparlistfalse
    \@topsep \@topsepadd
  \fi
  \advance\@topsep \parskip
  \leftskip \z@skip
  \rightskip \@rightskip
  \parfillskip \@flushglue
  \par@deathcycles \z@
  \@setpar{\if@newlist
             \advance\par@deathcycles \@ne
             \ifnum \par@deathcycles >\@m
               \@noitemerr
               {\@@par}%
             \fi
           \else
             {\@@par}%
           \fi}%
  \global \@newlisttrue
  \@outerparskip \parskip}
```

`\trivlist`

```
\def\trivlist{%
  \parsep\parskip
  \@nmbrlistfalse
  \@trivlist
  \labelwidth\z@
  \leftmargin\z@
  \itemindent\z@
```

We initialise `\@itemlabel` so that a `trivlist` with an `\item` not having an optional argument doesn't produce an error message.

```
  \let\@itemlabel\@empty
  \def\makelabel##1{##1}}
```

`\endlist`

```
\def\endlist{%
```

```
  \global\advance\@listdepth\m@ne
  \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`

```
\def\endtrivlist{%
  \if@inlabel
    \leavevmode
    \global \@inlabelfalse
  \fi
  \if@newlist
    \@noitemerr
    \global \@newlistfalse
  \fi
  \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.

```
  \else
    \@inmatherr{\end{\@currenvir}}%
  \fi
  \if@noparlist \else
    \ifdim\lastskip >\z@
      \@tempskipa\lastskip \vskip -\lastskip
      \advance\@tempskipa\parskip \advance\@tempskipa -\@outerparskip
      \vskip\@tempskipa
    \fi
    \@endparenv
  \fi
}
```

`\@endparenv` `\@doendpe` To suppress the paragraph indentation in text immediately following a paragraph-making environment, `\everypar` is changed to remove the space, and `\par` is redefined to restore `\everypar`. Instead of redefining `\par` and `\everpar`, `\@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)

```
\def\@endparenv{%
  \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}
\def\@doendpe{\@endpetrue
     \def\par{\@restorepar\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).

```
               {{\setbox\z@\lastbox}\everypar{}\@endpefalse}}
```

`\if@endpe`
`\@endpefalse`
`\@endpeltrue`

```
\newif\if@endpe
\@endpefalse
```

`\@mklab`

```
\def\@mklab#1{\hfil #1}
```

`\item`

```
\def\item{%
  \@inmatherr\item
  \@ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

`\@donoparitem`

```
\def\@donoparitem{%
  \@noparitemfalse
  \global\setbox\@labels\hbox{\hskip -\leftmargin
                              \unhbox\@labels
                               \hskip \leftmargin}%
  \if@minipage\else
    \@tempskipa\lastskip
    \vskip -\lastskip
    \advance\@tempskipa\@outerparskip
    \advance\@tempskipa -\parskip
    \vskip\@tempskipa
  \fi}
```

`\@item`

```
\def\@item[#1]{%
  \if@noparitem
    \@donoparitem
  \else
    \if@inlabel
      \indent \par
    \fi
    \ifhmode
      \unskip\unskip \par
    \fi
    \if@newlist
      \if@nobreak
        \@nbitem
      \else
        \addpenalty\@beginparpenalty
        \addvspace\@topsep
        \addvspace{-\parskip}%
      \fi
    \else
      \addpenalty\@itempenalty
      \addvspace\itemsep
    \fi
    \global\@inlabeltrue
  \fi
  \everypar{%
    \@minipagefalse
    \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.

```
    \if@inlabel
      \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.

```
      {\setbox\z@\lastbox
       \ifvoid\z@
          \kern-\itemindent
       \fi}%
      \box\@labels
      \penalty\z@
    \fi
```

This code is intended to prevent a page break after the first line of an item that comes immediately after a section title. It may be sensible to always forbid a page break after one line of an item? As with all such settings of `\clubpenalty` it is local so will have no effect if the item starts in a group.

Only resetting `\@nobreak` when it is true is now essential since now it is sometimes set locally.

```
    \if@nobreak
      \@nobreakfalse
      \clubpenalty \@M
    \else
      \clubpenalty \@clubpenalty
      \everypar{}%
    \fi}%
```

```
  \if@noitemarg
    \@noitemargfalse
    \if@nmbrlist
```

```
      \refstepcounter\@listctr
    \fi
  \fi
```

We use `\sbox` to support colour commands.

```
  \sbox\@tempboxa{\makelabel{#1}}%
  \global\setbox\@labels\hbox{%
    \unhbox\@labels
    \hskip \itemindent
    \hskip -\labelwidth
    \hskip -\labelsep
    \ifdim \wd\@tempboxa >\labelwidth
      \box\@tempboxa
```

```
    \else
      \hbox to\labelwidth {\unhbox\@tempboxa}%
    \fi
    \hskip \labelsep}%
  \ignorespaces}
```

`\makelabel`

```
\def\makelabel#1{%
  \@latex@error{Lonely \string\item--perhaps a missing
        list environment}\@ehc}
```

`\@nbitem`

```
\def\@nbitem{%
  \@tempskipa\@outerparskip
  \advance\@tempskipa -\parskip
  \addvspace\@tempskipa}
```

`\usecounter`

```
\def\usecounter#1{\@nmbrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}
```

56.6 Itemize and Enumerate

Enumeration is done with four counters: `enumi`, `enumii`, `enumiii` and `enumiv`, where `enum`N 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@enum`N`\theenum`N defines the output of a `\ref` command. A typical definition might be:

```
\def\theenumii{\alph{enumii}}
\def\p@enumii{\theenumi}
\def\labelenumii{(\theenumii)}
```

which will print the labels as '(a)', '(b)', ... and print a `\ref` as '3a'.

The item numbers are moved to the right of the label box, so they are always a distance of `\labelsep` from the item.

`\@enumdepth` holds the current enumeration nesting depth.

Itemization is controlled by four commands: `\labelitemi`, `\labelitemii`, `\labelitemiii`, and `\labelitemiv`. To cause the second-level list to be bulleted, you just define `\labelitemii` to be •. `\@itemspacing` and `\@itemdepth` are the analogs of `\@enumspacing` and `\@enumdepth`.

```
\enumerate ==
  BEGIN
    if \@enumdepth > 3
      then errormessage: "Too deeply nested".
      else \@enumdepth :=L \@enumdepth + 1
           \@enumctr :=L eval(enum@\romannumeral\the\@enumdepth)
           \list{\label(\@enumctr)}
                {\usecounter{\@enumctr}
                 \makelabel{LABEL} == \hss \llap{LABEL}}
    fi
  END

\endenumerate == \endlist
```

`\@enumdepth`

```
\newcount\@enumdepth \@enumdepth = 0
```

`\c@enumi`
`\c@enumii`
`\c@enumii`
`\c@enumiv`

```
\@definecounter{enumi}
\@definecounter{enumii}
\@definecounter{enumiii}
\@definecounter{enumiv}
```

`enumerate`

```
\def\enumerate{%
  \ifnum \@enumdepth >\thr@@\@toodeep\else
    \advance\@enumdepth\@ne
    \edef\@enumctr{enum\romannumeral\the\@enumdepth}%

      \expandafter
      \list
        \csname label\@enumctr\endcsname
        {\usecounter\@enumctr\def\makelabel##1{\hss\llap{##1}}}%
  \fi}
\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`

```
\newcount\@itemdepth \@itemdepth = 0
```

`itemize`

```
\def\itemize{%
  \ifnum \@itemdepth >\thr@@\@toodeep\else
    \advance\@itemdepth\@ne
    \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%

    \expandafter
    \list
      \csname\@itemitem\endcsname
      {\def\makelabel##1{\hss\llap{##1}}}%
  \fi}

\let\enditemize =\endlist
⟨/2ekernel⟩
```

File B

ltboxes.dtx

57 LaTeX Box commands

`\makebox` \makebox[⟨*wid*⟩][⟨*pos*⟩]{⟨*obj*⟩}
Puts ⟨*obj*⟩ in an `\hbox` of width ⟨*wid*⟩, positioned by ⟨*pos*⟩.

The possible ⟨*pos*⟩ are:

- `s` stretched,
- `l` flushleft,
- `r` flushright,
- `c` (default) centred.

If ⟨*wid*⟩ is missing, then ⟨*pos*⟩ is also missing and ⟨*obj*⟩ is put in an `\hbox` of its natural width.

\makebox(⟨*x*⟩,⟨*y*⟩)[⟨*pos*⟩]{⟨*obj*⟩}
Puts ⟨*obj*⟩ in an `\hbox` of width $x*$`\unitlength` and height $y*$`\unitlength`. ⟨*pos*⟩ 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 `\makebox` a [b] aligns on the *bottom* of the text as documented. If you want to align on the *baseline* use `\makebox( , )[b]{\raisebox{0pt}[\height][0pt]{xyz}}` or `\makebox( , )[b]{\smash{xyz}}`

`\mbox` \mbox{⟨*obj*⟩} The same as \makebox{⟨*obj*⟩}, but is more efficient as no checking for optional arguments is done.

`\newsavebox` `\newsavebox{\cmd}` : If `\cmd` is undefined, then defines it to be a TeX box register.

`\savebox` `\savebox{\cmd}` ... : `\cmd` is defined to be a TeX box register, and the '...' are any `\makebox` arguments. It is like `\makebox`, except it doesn't produce text but saves the value in `\box \cmd`.

`\sbox` \sbox{⟨*cmd*⟩}{⟨*obj*⟩} is an efficient abbreviation for
\savebox{⟨*cmd*⟩}{⟨*obj*⟩}.

`lrbox` \begin{lrbox}{⟨*cmd*⟩}⟨*text*⟩\end{lrbox} is equivalent to
\sbox{⟨*cmd*⟩}{⟨*text*⟩}
except that any white space at the beginning and end of ⟨*text*⟩ is ignored.

`\framebox` `\framebox` ... : like `\makebox`, except it puts a 'frame' around the box. The frame is made of lines of thickness `\fboxrule`, separated by space `\fboxsep` from the text – except for `\framebox(X,Y)` ... , where the thickness of the lines is as for the picture environment, and there is no separation added.

`\fbox` \fbox{⟨*obj*⟩} is an abbreviation for \framebox{⟨*obj*⟩}.

`\parbox` \parbox[⟨*pos*⟩][⟨*height*⟩][⟨*inner-pos*⟩]{⟨*width*⟩}{⟨*text*⟩} : Makes a box with `\hsize` ⟨*width*⟩, positioned by ⟨*pos*⟩ as follows: c : `\vcenter` (placed in `$...$` if not in math mode) b : `\vbox` t : `\vtop` default value is c. Sets `\hsize` := ⟨*width*⟩ and calls `\@parboxrestore`, which does the following: Restores the original definitions of:

```
\par
\\
\-
\'
 \`
\=
```

Resets the following parameters:

`\parindent` = 0pt
`\parskip` = 0pt   added 20 Jan 87
`\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 restore `\\`.

`minipage`

`minipage` : Similar to `\parbox`, except it also makes this look like a page by setting

`\textwidth` == `\columnwidth` == box width

changes footnotes by redefining:

`\@mpfn` == mpfootnote
`\thempfn` == `\thempfootnote`
`\@footnotetext` == `\@mpfootnotetext`

resets the following list environment parameters:

`\@listdepth` == `\@mplistdepth`
where `\@mplistdepth` is initialized to zero,

and executes `\@minipagerestore` to allow the document style to reset any other parameters it desires. It sets `@minipage` true, and resets `\everypar` to set it false. This switch keeps `\addvspace` from putting space at the top of a minipage.

Change added 24 May 89: `\minipage` sets `@minipage` globally; `\endminipage` resets it false.

`\rule`

`\rule[`⟨*raised*⟩`]{`⟨*width*⟩`}{`⟨*height*⟩`}` : Makes a ⟨*width*⟩ ∗ ⟨*height*⟩ rule, raised ⟨*raised*⟩.

`\underline`

`\underline{`⟨*text*⟩`}` : Makes an underlined hbox with ⟨*text*⟩ in it.

`\raisebox`

`\raisebox{`⟨*distance*⟩`}[`⟨*height*⟩`][`⟨*depth*⟩`]{`⟨*box*⟩`}` :
Raises ⟨*box*⟩ up by ⟨*distance*⟩ length (down if ⟨*distance*⟩ negative). Makes TeX think that the new box extends ⟨*height*⟩ above the line and ⟨*depth*⟩ below, for a total vertical length of ⟨*height*⟩+⟨*depth*⟩. Default values of ⟨*height*⟩ & ⟨*depth*⟩ = actual height and depth of box in new position.

```
⟨*2ekernel⟩
\message{boxes,}
```

`\makebox`

`\makebox` User level command just looks for optional `[` or `(`.

```
\def\makebox{%
  \leavevmode
  \@ifnextchar(%)
    \@makepicbox
    {\@ifnextchar[\@makebox\mbox}}
```

`\mbox`

The basic horizontal box command for LaTeX.

```
\long\def\mbox#1{\leavevmode\hbox{#1}}
```

`\@makebox`

Look for a possible second optional argument (defaults to `c`).

```
\def\@makebox[#1]{%
  \@ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}
```

`\@begin@tempboxa`

Helper macro for supporting `\height`, `\width` etc. Grab `#1` into `\@tempboxa` and measure it.

```
\long\def\@begin@tempboxa#1#2{%
   \begingroup
     \setbox\@tempboxa#1{\color@begingroup#2\color@endgroup}%
     \def\width{\wd\@tempboxa}%
     \def\height{\ht\@tempboxa}%
     \def\depth{\dp\@tempboxa}%
     \let\totalheight\@ovri
     \totalheight\height
     \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.

```
\let\@end@tempboxa\endgroup
```

`\bm@c`
`\bm@l`
`\bm@r`
`\bm@s`
`\bm@t`
`\bm@b`

Set up spacing.

```
\def\bm@c{\hss\unhbox\@tempboxa\hss}
\def\bm@l{\unhbox\@tempboxa\hss}\let\bm@t\bm@l
\def\bm@r{\hss\unhbox\@tempboxa}\let\bm@b\bm@r
\def\bm@s{\unhbox\@tempboxa}
```

`\@imakebox`

Internal form of `\makebox`.

```
\long\def\@imakebox[#1][#2]#3{%
  \@begin@tempboxa\hbox{#3}%
    \setlength\@tempdima{#1}%       support calc
    \hb@xt@\@tempdima{\csname bm@#2\endcsname}%
  \@end@tempboxa}
```

`\@makepicbox`

Picture mode form of `\makebox`.

```
\def\@makepicbox(#1,#2){%
  \@ifnextchar[{\@imakepicbox(#1,#2)}{\@imakepicbox(#1,#2)[]}}
```

`\@imakepicbox`

picture mode version

```
\long\def\@imakepicbox(#1,#2)[#3]#4{%
  \vbox to#2\unitlength
   {\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
    \let\mb@t\vss
    \@tfor\reserved@a :=#3\do{%
      \if s\reserved@a
        \let\mb@l\relax\let\mb@r\relax
      \else
        \expandafter\let\csname mb@\reserved@a\endcsname\relax
      \fi}%
    \mb@t
    \hb@xt@ #1\unitlength{\mb@l #4\mb@r}%
    \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.

```
    \kern\z@}}
```

`\set@color`

This macro is initialy a no-op, but the colour package will redefine it to insert a `\special`.

```
\let\set@color\relax
```

`\color@begingroup`
`\color@endgroup`
`\color@setgroup`
`\normalcolor`
`\color@hbox`
`\color@vbox`
`\color@endbox`

These macros are initialy a no-op, but the colour package will redefine them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`, `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and ⟨*set to main document colour*⟩ respectively.

```
\let\color@begingroup\relax
\let\color@endgroup\relax
\let\color@setgroup\relax
\let\normalcolor\relax
\let\color@hbox\relax
\let\color@vbox\relax
\let\color@endbox\relax
```

`\newsavebox` Allocate a new ‘savebox’.

```
\def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}
```

`\savebox` Save `#1` in a box register.

```
\def\savebox#1{%
  \@ifnextchar(%)
    {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}
```

`\sbox` Save `#1` in a box register.

```
\long\def\sbox#1#2{\setbox#1\hbox{%
  \color@setgroup#2\color@endgroup}}
```

`\@savebox` Look for second optional argument.

```
\def\@savebox#1[#2]{%
  \@ifnextchar [{\@isavebox#1[#2]}{\@isavebox#1[#2][c]}}
```

`\@isavebox`

```
\long\def\@isavebox#1[#2][#3]#4{%
  \sbox#1{\@imakebox[#2][#3]{#4}}}
```

`\@savepicbox` Picture mode version of `\savebox`.

```
\def\@savepicbox#1(#2,#3){%
  \@ifnextchar[%]
    {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3)[]}}
```

`\@isavepicbox` Picture mode version of `\savebox`.

```
\long\def\@isavepicbox#1(#2,#3)[#4]#5{%
  \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.

```
\def\lrbox#1{%
  \edef\reserved@a{%
    \endgroup
    \setbox#1\hbox{%
      \begingroup\aftergroup}%
        \def\noexpand\@currenvir{\@currenvir}%
        \def\noexpand\@currenvline{\on@line}}%
  \reserved@a
    \@endpefalse
    \color@setgroup
      \ignorespaces}
```

`\endlrbox` End the `lrbox` environment.

```
\def\endlrbox{\unskip\color@endgroup}
```

`\usebox` unchanged

```
\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 occurence of `\@halfwidth` by `\@wholewidth`

```
\long\def\frame#1{%
  \leavevmode
  \hbox{%
    \hskip-\@wholewidth
    \vbox{%
      \vskip-\@wholewidth
      \hrule \@height\@wholewidth
      \hbox{%
        \vrule\@width\@wholewidth
        #1%
        \vrule\@width\@wholewidth}%
      \hrule\@height\@wholewidth
      \vskip-\@wholewidth}%
    \hskip-\@wholewidth}}
```

`\fboxrule` `\fboxsep` user level parameters,

```
\newdimen\fboxrule
\newdimen\fboxsep
```

`\fbox` Abbreviated framed box command.

```
\long\def\fbox#1{%
  \leavevmode
  \setbox\@tempboxa\hbox{%
    \color@begingroup
      \kern\fboxsep{#1}\kern\fboxsep
    \color@endgroup}%
  \@frameb@x\relax}
```

`\framebox` Framed version of `\makebox`.

```
\def\framebox{%
  \@ifnextchar(%)
    \@framepicbox{\@ifnextchar[\@framebox\fbox}}
```

`\@framebox` Deal with optional arguments.

```
\def\@framebox[#1]{%
  \@ifnextchar[%]
    {\@iframebox[#1]}%
    {\@iframebox[#1][c]}}
```

`\@iframebox` The handling the optional arguments. In order to set the whole box, including the frame to the specified dimension, we first determine that dimension from the natural size of the text, `#3`. calculated width.

```
\long\def\@iframebox[#1][#2]#3{%
  \leavevmode
  \@begin@tempboxa\hbox{#3}%
  \setlength\@tempdima{#1}%
  \setbox\@tempboxa\hb@xt@\@tempdima
         {\kern\fboxsep\csname bm@#2\endcsname\kern\fboxsep}%
  \@frameb@x{\kern-\fboxrule}%
  \@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.

```
\def\@frameb@x#1{%
  \@tempdima\fboxrule
  \advance\@tempdima\fboxsep
  \advance\@tempdima\dp\@tempboxa
  \hbox{%
    \lower\@tempdima\hbox{%
      \vbox{%
        \hrule\@height\fboxrule
        \hbox{%
          \vrule\@width\fboxrule
          #1%
          \vbox{%
            \vskip\fboxsep
            \box\@tempboxa
            \vskip\fboxsep}%
          #1%
          \vrule\@width\fboxrule}%
        \hrule\@height\fboxrule}%
                          }%
        }%
}
```

`\@framepicbox` Picture mode version.

```
\def\@framepicbox(#1,#2){%
  \@ifnextchar[{\@iframepicbox(#1,#2)}{\@iframepicbox(#1,#2)[]}}
```

`\@iframepicbox` Picture mode version.

```
\long\def\@iframepicbox(#1,#2)[#3]#4{%
  \frame{\@imakepicbox(#1,#2)[#3]{#4}}}
```

`\parbox` The main vertical-box command for LaTeX.

```
\def\parbox{%
  \@ifnextchar[%]
    \@iparbox
    {\@iiiparbox c\relax[s]}}
```

`\@iparbox` Optional argument handling.

```
\def\@iparbox[#1]{%
  \@ifnextchar[%]
    {\@iiparbox{#1}}%
    {\@iiiparbox{#1}\relax[s]}}
```

`\@iiparbox` Optional argument handling.

```
\def\@iiparbox#1[#2]{%
  \@ifnextchar[%]
    {\@iiiparbox{#1}{#2}}%
    {\@iiiparbox{#1}{#2}[#1]}}
```

`\@iiiparbox` The internal version of `\parbox`.

`\@parboxto`

```
\let\@parboxto\@empty
\long\def\@iiiparbox#1#2[#3]#4#5{%
  \leavevmode
  \@pboxswfalse
  \setlength\@tempdima{#4}%
  \@begin@tempboxa\vbox{\hsize\@tempdima\@parboxrestore#5\@@par}%
    \ifx\relax#2\else
      \setlength\@tempdimb{#2}%
      \edef\@parboxto{to\the\@tempdimb}%
    \fi
    \if#1b\vbox
    \else\if #1t\vtop
    \else\ifmmode\vcenter
    \else\@pboxswtrue $\vcenter
    \fi\fi\fi
    \@parboxto{\let\hss\vss\let\unhbox\unvbox
       \csname bm@#3\endcsname}%
    \if@pboxsw \m@th$\fi
  \@end@tempboxa}
```

`\@arrayparboxrestore` Restore various paragraph parameters.

The rational 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.

```
\def\@arrayparboxrestore{%
  \let\if@nobreak\iffalse
  \let\if@noskipsec\iffalse
  \let\par\@@par
  \let\-\@dischyph
```

Redefined accents to allow changes in font encoding

```
  \let\'\@acci\let\`\@accii\let\=\@acciii
  \parindent\z@ \parskip\z@skip
  \everypar{}%
  \linewidth\hsize
  \@totalleftmargin\z@
  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
  \parfillskip\@flushglue \lineskip\normallineskip
  \baselineskip\normalbaselineskip
  \sloppy}
```

`\parboxrestore` Restore various paragraph parameters, and also `\\`.

```
\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
```

`\if@minipage` Switch that is true at the start of a minipage.

```
\def\@minipagefalse{\global\let\if@minipage\iffalse}
\def\@minipagetrue {\global\let\if@minipage\iftrue}
\@minipagefalse
```

`\minipage` Essentially an environment form of `\parbox`.

```
\def\minipage{%
  \@ifnextchar[%]
    \@iminipage
    {\@iiiminipage c\relax[s]}}
```

`\@iminipage` Optional argument handling.

```
\def\@iminipage[#1]{%
  \@ifnextchar[%]
    {\@iiminipage{#1}}%
    {\@iiiminipage{#1}\relax[s]}}
```

`\@iiminipage` Optional argument handling.

```
\def\@iiminipage#1[#2]{%
  \@ifnextchar[%]
    {\@iiiminipage{#1}{#2}}%
    {\@iiiminipage{#1}{#2}[#1]}}
```

`\@iiiminipage` Internal form of `minipage`.

```
\def\@iiiminipage#1#2[#3]#4{%
  \leavevmode
  \@pboxswfalse
  \setlength\@tempdima{#4}%
  \def\@mpargs{{#1}{#2}[#3]{#4}}%
  \setbox\@tempboxa\vbox\bgroup
    \color@begingroup
      \hsize\@tempdima
      \textwidth\hsize \columnwidth\hsize
      \@parboxrestore
      \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
      \let\@footnotetext\@mpfootnotetext
      \let\@listdepth\@mplistdepth \@mplistdepth\z@
      \@minipagerestore
      \@setminipage}
```

`\@minipagerestore` Hook so that other styles can reset other commands in a minipage.

```
 \let\@minipagerestore=\relax
```

`\endminipage`

```
\def\endminipage{%
    \par
    \unskip
    \ifvoid\@mpfootins\else
      \vskip\skip\@mpfootins
      \normalcolor
      \footnoterule
      \unvbox\@mpfootins
    \fi
    \@minipagefalse   %% added 24 May 89
  \color@endgroup
  \egroup
  \expandafter\@iiiparbox\@mpargs{\unvbox\@tempboxa}}
```

`\@mplistdepth` `\@mpfootins` Versions of `\@listdepth` and `\footins` local to minipage.

```
\newcount\@mplistdepth
\newinsert\@mpfootins
```

`\@mpfootnotetext` Minipage version of `\@footnotetext`.

Final `\strut` added 27 Mar 89, on suggestion by Don Hosek

```
\long\def\@mpfootnotetext#1{%
  \global\setbox\@mpfootins\vbox{%
    \unvbox\@mpfootins
    \reset@font\footnotesize
    \hsize\columnwidth
    \@parboxrestore
    \protected@edef\@currentlabel
         {\csname p@mpfootnote\endcsname\@thefnmark}%
    \color@begingroup
      \@makefntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
    \color@endgroup}}
```

```
\newif\if@pboxsw
```

`\rule` Draw a rule of the specified size.

```
\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}
```

`\@rule` Internal form of `\rule`.

```
 \def\@rule[#1]#2#3{%
   \leavevmode
   \hbox{%
     \setlength\@tempdima{#1}%
     \setlength\@tempdimb{#2}%
     \setlength\@tempdimc{#3}%
     \advance\@tempdimc\@tempdima
     \vrule\@width\@tempdimb\@height\@tempdimc\@depth-\@tempdima}}
```

`\@@underline` Saved primitive `\underline`.

```
\let\@@underline\underline
```

`\underline` LaTeX version works outside math.

```
\def\underline#1{%
  \relax
  \ifmmode\@@underline{#1}%
  \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}
```

`\raisebox` Raise a box, and change its vertical dimensions.

```
\def\raisebox#1{%
  \leavevmode
  \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
```

`\@rsbox` Optional argument handling.

```
\def\@rsbox#1[#2]{%
  \@ifnextchar[{\@iirsbox{#1}[#2]}{\@irsbox{#1}[#2]}}
```

`\@argrsbox` …

`\@irsbox` Internal version of `\raisebox` (less than two optional args).

```
\long\def\@irsbox#1[#2]#3{%
  \@begin@tempboxa\hbox{#3}%
    \setlength\@tempdima{#1}%
    \ifx\\#2\\\else\setlength\@tempdimb{#2}\fi
    \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
    \ifx\\#2\\\else\ht\@tempboxa\@tempdimb\fi
    \box\@tempboxa
  \@end@tempboxa}
```

`\@iirsbox` Internal version of `\raisebox` (two optional args).

```
\long\def\@iirsbox#1[#2][#3]#4{%
  \@begin@tempboxa\hbox{#4}%
    \setlength\@tempdima{#1}%
    \setlength\@tempdimb{#2}%
    \setlength\dimen@{#3}%
    \setbox\@tempboxa\hbox{\raise\@tempdima\box\@tempboxa}%
    \ht\@tempboxa\@tempdimb
    \dp\@tempboxa\dimen@
    \box\@tempboxa
  \@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.

```
\def\@finalstrut#1{%
  \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}
```

57.1 Some low-level constructs

The following commands are basically inherited from plain TeX.

`\leftline` `\rightline` `\centerline` `\@@line` These macros place text on a full line either centred or left or right adjusted.

```
\def\@@line{\hb@xt@\hsize}
\def\leftline#1{\@@line{#1\hss}}
\def\rightline#1{\@@line{\hss#1}}
\def\centerline#1{\@@line{\hss#1\hss}}
```

`\rlap` `\llap` These macros place text to the left or right of the current reference point without taking up space.

```
\def\rlap#1{\hb@xt@\z@{#1\hss}}
\def\llap#1{\hb@xt@\z@{\hss#1}}
```

```
⟨/2ekernel⟩
```

File C
lttab.dtx

58 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 LaTeX 2.09 version, not the extended version described in *The LaTeX Companion*. Use the `array` package to obtain the extended version.

58.1 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 untabs”        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 untabs”
       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.

```
⟨2ekernel⟩\expandafter\let\csname ver@autotabg.sty\endcsname\fmtversion
```

`\@firsttab`
`\@maxtab`

```
⟨*2ekernel | autoload⟩
\newdimen\@gtempa
\chardef\@firsttab=\the\allocationnumber
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa\newdimen\@gtempa
\newdimen\@gtempa
\chardef\@maxtab=\the\allocationnumber
\dimen\@firsttab=0pt
```

`\@nxttabmar`
`\@curtabmar`
`\@curtab`
`\@hightab`
`\@tabpush`

```
\newcount\@nxttabmar
\newcount\@curtabmar
\newcount\@curtab
\newcount\@hightab
\newcount\@tabpush
```

`\@curline`
`\@curfield`
`\@tabfbox`

```
\newbox\@curline
\newbox\@curfield
\newbox\@tabfbox
```

```
⟨/2ekernel | autoload⟩
⟨*2ekernel | def⟩
```

`\if@rjfield`

```
\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.

```
\gdef\@startline{%
     \ifnum \@nxttabmar >\@hightab
       \@badtab
       \global\@nxttabmar \@hightab
     \fi
     \global\@curtabmar \@nxttabmar
     \global\@curtab \@curtabmar
     \global\setbox\@curline \hbox {}%
     \@startfield
     \strut}
```

`\@stopline`

```
\gdef\@stopline{%
 \unskip
 \@stopfield
 \if@rjfield
   \global\@rjfieldfalse
   \@tempdima\@totalleftmargin
   \advance\@tempdima\linewidth
   \hb@xt@\@tempdima{%
     \@itemfudge\hskip\dimen\@curtabmar
     \box\@curline
     \hfil
     \box\@curfield}%
 \else
   \@addfield
   \hbox{\@itemfudge\hskip\dimen\@curtabmar\box\@curline}%
 \fi}
```

`\@startfield`

```
\gdef\@startfield{%
 \global\setbox\@curfield\hbox\bgroup\color@begingroup}
```

`\@stopfield`

```
\gdef\@stopfield{%
 \color@endgroup\egroup}
```

`\@contfield`

```
\gdef\@contfield{%
 \global\setbox\@curfield\hbox\bgroup\color@begingroup
 \unhbox\@curfield}
```

`\@addfield`

```
\gdef\@addfield{\global\setbox\@curline\hbox{\unhbox
     \@curline\unhbox\@curfield}}
```

`\@ifatmargin`

```
\gdef\@ifatmargin{\ifdim \wd\@curline =\z@}
```

`\@tabcr`

```
\gdef\@tabcr{\@stopline \@ifstar{\penalty \@M \@xtabcr}\@xtabcr}
```

`\@xtabcr`

```
\gdef\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}
```

`\@itabcr`

```
\gdef\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
```

```
\gdef\kill{\@stopfield\@startline\ignorespaces}
```

`\tabbing` We use `\relax` to prevent `\item` from scanning too far.

```
\gdef\tabbing{\lineskip \z@skip\let\>\@rtab\let\<\@ltab\let\=\@settab
     \let\+\@tabplus\let\-\@tabminus\let\`\@tabrj\let\'\@tablab
     \let\\=\@tabcr
     \@hightab\@firsttab
     \global\@nxttabmar\@firsttab
     \dimen\@firsttab\@totalleftmargin
     \global\@tabpush\z@ \global\@rjfieldfalse
     \trivlist \item\relax
     \if@minipage\else\vskip\parskip\fi
```

```
     \setbox\@tabfbox\hbox{%
       \rlap{\hskip\@totalleftmargin\indent\the\everypar}}%
     \def\@itemfudge{\box\@tabfbox}%
     \@startline\ignorespaces}
```

`\endtabbing`

```
\gdef\endtabbing{%
 \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}
```

`\@rtab` Omitted `\global` added to `\@rtab` 17 Jun 86

```
\gdef\@rtab{\@stopfield\@addfield\ifnum \@curtab<\@hightab
      \global\advance\@curtab \@ne \else\@badtab\fi
      \@tempdima\dimen\@curtab
      \advance\@tempdima -\dimen\@curtabmar
      \advance\@tempdima -\wd\@curline
      \global\setbox\@curline\hbox{\unhbox\@curline\hskip\@tempdima}%
      \@startfield\ignorespaces}
```

`\@settab`

```
\gdef\@settab{\@stopfield\@addfield
  \ifnum \@curtab <\@maxtab
    \ifnum\@curtab =\@hightab
      \advance\@hightab \@ne
    \fi
    \global\advance\@curtab \@ne
  \else
    \@latex@error{Tab overflow}\@ehd
  \fi
  \dimen\@curtab \dimen\@curtabmar
  \advance\dimen\@curtab \wd\@curline
  \@startfield
  \ignorespaces}
```

`\@ltab`

```
\gdef\@ltab{\@ifatmargin\ifnum\@curtabmar >\@firsttab
      \global\advance\@curtab \m@ne \global\advance\@curtabmar\m@ne\else
      \@badtab\fi\else
      \@latex@error{\string\<\space in mid line}\@ehd\fi\ignorespaces}
```

`\@tabplus`

```
\gdef\@tabplus{%
  \ifnum\@nxttabmar<\@hightab
    \global\advance\@nxttabmar\@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
```

`\@tabminus`

```
\gdef\@tabminus{%
  \ifnum\@nxttabmar>\@firsttab
    \global\advance\@nxttabmar\m@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
```

`\@tabrj`

```
\gdef\@tabrj{%
  \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}
```

`\@tablab` `\setbox\@curline` made `\global` in `\@tablab`. 17 Jun 86

```
\gdef\@tablab{%
  \@stopfield
  \global\setbox\@curline\hbox{%
    \box\@curline
    \hskip-\wd\@curfield \hskip-\tabbingsep
    \box\@curfield
    \hskip\tabbingsep}%
  \@startfield
  \ignorespaces}
```

`\pushtabs`

```
\gdef\pushtabs{%
  \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
       \@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.

```
\gdef\poptabs{\@stopfield\@addfield
  \ifnum \@tabpush >\z@
    \endgroup
    \global\advance\@tabpush \m@ne
    \ifnum \@curtab >\@hightab
      \global \@curtab \@hightab
      \@badtab
    \fi
  \else
    \@badpoptabs
  \fi
  \@contfield}
```

```
⟨/2ekernel | def⟩
```

`\tabbingsep`

```
⟨*2ekernel | autoload⟩
\newdimen\tabbingsep
⟨/2ekernel | autoload⟩
```

`\tabbing`

```
⟨*autoload⟩
\def\tabbing{\@autoload{tabg}\tabbing}
⟨/autoload⟩
```

58.2 array and tabular environments

ARRAY PARMETERS:
`\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` commmand.

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  \crcr }}  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 \crcr}} $} 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 $
 END

\@argarraycr[LENGTH] ==
 BEGIN
   $                      %% Needed to balance $ of \@arraycr
   if LENGTH > 0
     then  \@tempdima := depth of \@arstrutbox + LENGTH
           \vrule height 0pt width 0pt depth \@tempdima
           \cr
     else  \cr \noalign{\vskip LENGTH}
 END

\@tabularcr and \@argtabularcr  same as \@arraycr and
\@argarraycr
  except without the extra $'s.
```

```
⟨*2ekernel | autoload⟩
```

`\extracolsep`

```
\def\extracolsep#1{\tabskip #1\relax}
```

`\array`

```
\def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
 \let\@classiv\@arrayclassiv
 \let\\\@arraycr\let\@halignto\@empty\@tabarray}
```

`\endarray`
`\endtabular`
`\endtabular*`

```
\def\endarray{\crcr\egroup\egroup}
\def\endtabular{\crcr\egroup\egroup $\egroup}
\expandafter \let \csname endtabular*\endcsname = \endtabular
```

`\tabular`

```
\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.

```
\@namedef{tabular*}#1{%
 \setlength\dimen@{#1}%
   \edef\@halignto{to\the\dimen@}\@tabular}
```

`\@tabular`

```
\def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
   \let\@classz\@tabclassz
   \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}
```

`\@tabarray` RmS 91/11/04 added `\m@th`.

```
\def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}
```

RmS 1993/11/03 changed `\halign` to `\ialign` and removed superfluous `\tabskip` assignment

`\@array`

```
\def\@array[#1]#2{%
  \if #1t\vtop \else \if#1b\vbox \else \vcenter \fi\fi
```

```
  \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.

```
  \setbox\@arstrutbox\hbox{%
    \vrule \@height\arraystretch\ht\strutbox
          \@depth\arraystretch \dp\strutbox
          \@width\z@}%
  \@mkpream{#2}%
  \edef\@preamble{%
    \ialign \noexpand\@halignto
      \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.

```
  \let\@startpbox\@@startpbox \let\@endpbox\@@endpbox
  \let\tabularnewline\\%
    \let\par\@empty
    \let\@sharp##%
    \set@typeset@protect
    \lineskip\z@skip\baselineskip\z@skip
```

If the parsing of the preamble goes wrong there my 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).

```
    \ifhmode \@preamerr\z@ \@@par\fi
    \@preamble}
```

`\@arraycr` Array version of `\\`.

```
\def\@arraycr{%
  ${\ifnum0=`}\fi\@ifstar\@xarraycr\@xarraycr}
```

`\@arraycr`

```
\def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0=`{\fi}${}\cr}}
```

`\@argarraycr`

```
\def\@argarraycr[#1]{%
  \ifnum0=`{\fi}${}\ifdim #1>\z@ \@xargarraycr{#1}\else
   \@yargarraycr{#1}\fi}
```

`\tabularnewline` Tabular version of `\\`.

```
\let\tabularnewline\relax
```

`\@tabularcr`

```
\def\@tabularcr{%
  {\ifnum0=`}\fi\@ifstar\@xtabularcr\@xtabularcr}
```

`\@xtabularcr`

```
\def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0=`{\fi}\cr}}
```

`\@argtabularcr`

```
\def\@argtabularcr[#1]{%
  \ifnum0=`{\fi}%
    \ifdim #1>\z@
      \unskip\@xargarraycr{#1}%
    \else
      \@yargarraycr{#1}%
    \fi}
```

`\@xargarraycr`

```
\def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
   \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}
```

`\@yargarraycr`

```
\def\@yargarraycr#1{\cr\noalign{\vskip #1}}
```

`\multicolumn`

```
\multicolumn{NUMBER}{FORMAT}{ITEM} ==
  BEGIN
  \multispan{NUMBER}
  \begingroup
  \@addamp == null
  \@mkpream{FORMAT}
  \@sharp == ITEM
  \protect == \relax
  \@startpbox == \@@startpbox
  \@endpbox == \@@endpbox
  \@arstrut
```

```
\@preamble
\endgroup
END
```

The command `\def\@addamp{}` was removed from `\multicolumn` on 6 Dec 86 because it caused embedded array environments not to work. I think that it was included originally to prevent an error message if the 2nd argument to the `\multicolumn` command had two column specifiers.

8 Feb 89 — `\hbox{}` added after `\@preamble` to correct bug that occurred if `\multicolumn` preceded `\\[D]` with `D > 0`, caused by `\\[]` command doing an `\unskip`, which removed `\tabcolsep` glue inserted by `\multicolumn`.

This has been made long so that, for example, a `p`-column can contain multiple paragraphs; maybe the arguments of `@`-expressions should also be able to contain multiple paragraphs.

```
\long\def\multicolumn#1#2#3{\multispan{#1}\begingroup
  \@mkpream{#2}%
  \def\@sharp{#3}\set@typeset@protect
  \let\@startpbox\@@startpbox\let\@endpbox\@@endpbox
  \@arstrut \@preamble\hbox{}\endgroup\ignorespaces}
```

Codes for classes and character numbers of array, tabular and multicolumn arguments.

Character	Class	Number
c	0	0
l	0	1
r	0	2
|	1	-
@	2	-
p	3	-
{@-exp}	4	-
{p-arg}	5	-

`\@testpach \foo` : expands `\foo`, which should be an array parameter token, and sets `\@chclass` and `\@chnum` to its class and number. Uses `\@lastchclass` to distinguish 4 and 5

Preamble error codes
0: 'illegal character'
1: 'Missing @-exp'
2: 'Missing p-arg'

```
\@addamp ==
  BEGIN if @firstamp = true then @firstamp := false
                              else &                      fi
  END

\@mkpream TOKENLIST ==
  BEGIN
   @firstamp     := T
   \@lastchclass := 6
   \@preamble    == null
   \@sharp       == \relax
   \protect      == BEGIN \noexpand\protect\noexpand END
   \@startpbox   == \relax
   \@endpbox     == \relax
   \@expast{TOKENLIST}
```

```
    for \@nextchar := expand(\reserved@a)
      do  \@testpach{\@nextchar}
          case of \@chclass
            0 -> \@classz
            1 -> \@classi
              ...
            5 -> \@classv
          end case
          \@lastchclass := \@chclass
      od
      case of \@lastchclass
         0 -> \hskip \arraycolsep             % lrc
         1 ->                                  % |
         2 -> \@preamerr1 % 'Missing @-exp'    % @
         3 -> \@preamerr2 % 'Missing p-arg'    % p
         4 ->                                  % @-exp
         5 -> \hskip \arraycolsep             % p-exp
      end case
   END

  \@arrayclassz ==
    BEGIN
      \@preamble := \@preamble *
              case of \@lastchclass
                 0 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
                 1 -> \@addamp \hskip \arraycolsep
                 2 ->  % impossible
                 3 ->  % impossible
                 4 -> \@addamp
                 5 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
                 6 -> \@addamp \hskip \arraycolsep
               end case
             * case of \@chnum
                  0 -> \hfil$\relax\@sharp$\hfil
                  1 -> $\relax\@sharp$\hfil
                  2 -> \hfil$\relax\@sharp$
               end case
    END

 \@tabclassz == similar to \@arrayclassz

 \@classi ==
  BEGIN
    \@preamble := \@preamble *
                 case of \@lastchclass
                    0 -> \hskip \arraycolsep \@arrayrule
                    1 -> \hskip \doublerulesep \@arrayrule
                    2 -> % impossible
                    3 -> % impossible
                    4 -> \@arrayrule
                    5 -> \hskip \arraycolsep \@arrayrule
                    6 -> \@arrayrule
                 end case
  END
 \@classii ==
  BEGIN
    \@preamble := \@preamble *
                  case of \@lastchclass
                     0    ->
                     1    -> \hskip .5\arrayrulewidth
                     2    -> % impossible
                     else ->
                  end case
  END

 \@classiii ==
  BEGIN
    \@preamble := \@preamble *
               case of \@lastchclass
                  0 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
                  1 -> \@addamp \hskip \arraycolsep
                  2 -> % impossible
                  3 -> % impossible
                  4 -> \@addamp
                  5 -> \hskip \arraycolsep \@addamp \hskip
\arraycolsep
                  6 -> \@addamp \hskip \arraycolsep
                end case
  END

 \@arrayclassiv  ==
      BEGIN  \@preamble := \@preamble * $ \@nextchar$  END

 \@tabclassiv   == same as \@arrayclassv except without the $ ... $

 \@classv ==
   BEGIN
    \@preamble :=
        \@preamble * \@startpbox{\@nextchar}\ignorespaces\@sharp
                                  \@endpbox
   END

 \@expast{S}:
  Sets \reserved@a := S with all instances of *{N}{STRING}
  replaced by N copies of STRING, where N > 0.  An *
  appearing inside braces is ignored, but *-expressions
  inside STRING are expanded, so nested *-expressions are
  handled properly.

 \@expast{S} == BEGIN  \@xexpast S *0x\@@  END

 \@xexpast S1 *{N}{S2} S3 \@@ ==
  BEGIN
    \reserved@a   := S1
    \@tempcnta := N
    if \@tempcnta > 0
      then  while \@tempcnta > 0 do \reserved@a   := \reserved@a S2
                                    \@tempcnta := \@tempcnta - 1 od
           \reserved@b == \@xexpast
      else  \reserved@b == \@xexnoop
  fi
  \expandafter \reserved@b \reserved@a S3 \@@
END
```

\@xexnoop

```
\def\@xexnoop #1\@@{}
```

\@expast

```
\def\@expast#1{\@xexpast #1*0x\@@}
```

\@xexpast

```
\def\@xexpast#1*#2#3#4\@@{%
  \edef\reserved@a{#1}%
  \@tempcnta#2\relax
  \ifnum\@tempcnta>\z@
    \@whilenum\@tempcnta>\z@\do
       {\edef\reserved@a{\reserved@a#3}\advance\@tempcnta \m@ne}%
    \let\reserved@b\@xexpast
  \else
    \let\reserved@b\@xexnoop
  \fi
  \expandafter\reserved@b\reserved@a #4\@@}
```

\if@firstamp

\@addamp

```
\newif\if@firstamp
\def\@addamp{%
  \if@firstamp
    \@firstampfalse
  \else
    \edef\@preamble{\@preamble &}%
  \fi}
```

\@arrayacol

\@tabacol

\@ampacol

\@acolampacol

```
\def\@arrayacol{\edef\@preamble{\@preamble \hskip \arraycolsep}}
\def\@tabacol{\edef\@preamble{\@preamble \hskip \tabcolsep}}
\def\@ampacol{\@addamp \@acol}
\def\@acolampacol{\@acol\@addamp\@acol}
```

\@mkpream

```
\def\@mkpream#1{\@firstamptrue\@lastchclass6
  \let\@preamble\@empty
  \let\protect\@unexpandable@protect
  \let\@sharp\relax
  \let\@startpbox\relax\let\@endpbox\relax
  \@expast{#1}%
  \expandafter\@tfor \expandafter
    \@nextchar \expandafter:\expandafter=\reserved@a\do
       {\@testpach\@nextchar
    \ifcase \@chclass \@classz \or \@classi \or \@classii \or \@classiii
      \or \@classiv \or\@classv \fi\@lastchclass\@chclass}%
  \ifcase \@lastchclass \@acol
      \or \or \@preamerr \@ne\or \@preamerr \tw@\or \or \@acol \fi}
```

\@arrayclassz

```
\def\@arrayclassz{\ifcase \@lastchclass \@acolampacol \or \@ampacol \or
   \or \or \@addamp \or
   \@acolampacol \or \@firstampfalse \@acol \fi
\edef\@preamble{\@preamble
  \ifcase \@chnum
     \hfil$\relax\@sharp$\hfil \or $\relax\@sharp$\hfil
    \or \hfil$\relax\@sharp$\fi}}
```

`\@tabclassz` RmS 91/08/14 inserted extra braces around entry for NFSS

```
\def\@tabclassz{%
  \ifcase\@lastchclass
    \@acolampacol
  \or
    \@ampacol
  \or
  \or
  \or
    \@addamp
  \or
    \@acolampacol
  \or
    \@firstampfalse\@acol
  \fi
  \edef\@preamble{%
    \@preamble{%
      \ifcase\@chnum
        \hfil\ignorespaces\@sharp\unskip\hfil
      \or
        \hskip1sp\ignorespaces\@sharp\unskip\hfil
      \or
        \hfil\hskip1sp\ignorespaces\@sharp\unskip
      \fi}}}
```

`\@classi`

```
\def\@classi{%
  \ifcase\@lastchclass
    \@acol\@arrayrule
  \or
    \@addtopreamble{\hskip \doublerulesep}\@arrayrule
  \or
  \or
  \or
    \@arrayrule
  \or
    \@acol\@arrayrule
  \or
    \@arrayrule
  \fi}
```

`\@classii`

```
\def\@classii{%
  \ifcase\@lastchclass
  \or
    \@addtopreamble{\hskip .5\arrayrulewidth}%
  \fi}
```

`\@classiii`

```
\def\@classiii{\ifcase \@lastchclass \@acolampacol \or
   \@addamp\@acol \or
   \or \or \@addamp \or
   \@acolampacol \or \@ampacol \fi}
```

`\@tabclassiv`

```
\def\@tabclassiv{\@addtopreamble\@nextchar}
```

`\@arrayclassiv`

```
\def\@arrayclassiv{\@addtopreamble{$\@nextchar$}}
```

\@classv

```
\def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
\@sharp\@endpbox}}
```

\@addtopreamble

```
\def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}
```

\@chclass
\@lastchclass
\@chnum

```
\newcount\@chclass
\newcount\@lastchclass
\newcount\@chnum
```

\arraycolsep
\tabcolsep
\arrayrulewidth
\doublerulesep

```
\newdimen\arraycolsep
\newdimen\tabcolsep
\newdimen\arrayrulewidth
\newdimen\doublerulesep
```

\arraystretch

```
\def\arraystretch{1}    % Default value.
```

\@arstrutbox
\@arstrut

```
\newbox\@arstrutbox
```

```
\def\@arstrut{%
  \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}
```

\@arrayrule

```
\def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
   \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}
```

\@testpatch

```
\def\@testpach#1{\@chclass \ifnum \@lastchclass=\tw@ 4 \else
    \ifnum \@lastchclass=3 5 \else
     \z@ \if #1c\@chnum \z@ \else
                              \if #1l\@chnum \@ne \else
                              \if #1r\@chnum \tw@ \else
          \@chclass \if #1|\@ne \else
                    \if #1@\tw@ \else
                    \if #1p3 \else \z@ \@preamerr 0\fi
  \fi  \fi  \fi  \fi  \fi  \fi
\fi}
```

\hline

```
\def\hline{%
  \noalign{\ifnum0=`}\fi\hrule \@height \arrayrulewidth \futurelet
   \reserved@a\@xhline}
```

\@xhline

```
\def\@xhline{\ifx\reserved@a\hline
              \vskip\doublerulesep
```

Measure from the middle of the rules.

```
              \vskip-\arrayrulewidth
             \fi
      \ifnum0=`{\fi}}
```

\vline

```
\def\vline{\vrule \@width \arrayrulewidth}
```

`\cline`
`\@cline`

The old LaTeX2.09 implementation of `\cline` used up quite a lot of memory and two precious count registers. This new (1995/09/14) implementation does not use any count registers. It is coded in a way that depends heavily on the definition of `\multispan` so that command has been moved here from the file `ltplain.dtx`.

These counters are no longer declared.

```
\newcount\@cla
\newcount\@clb
```

```
\def\cline#1{\@cline#1\@nil}
```

```
\def\@cline#1-#2\@nil{%
  \omit
```

Use the counter from `\multispan`.

```
  \@multicnt#1%
  \advance\@multispan\m@ne
  \ifnum\@multicnt=\@ne\@firstofone{&\omit}\fi
  \@multicnt#2%
  \advance\@multicnt-#1%
  \advance\@multispan\@ne
```

The original had `\unskip` at this point, but how could a skip get here ???

```
  \leaders\hrule\@height\arrayrulewidth\hfill
  \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.

```
  \noalign{\vskip-\arrayrulewidth}}
```

`\mscount`

The `\mscount` counter is no longer declared, saving a csname and a register. It is declared in compatibility mode.

`\multispan`
`\@multispan`
`\sp@n`

Modify `\multispan` slightly from its plain TeX definition to allow more efficient code sharing with `\multicolumn`. Also share a count register with `\multiput`.

```
\def\multispan{\omit\@multispan}
```

```
\def\@multispan#1{%
  \@multicnt#1\relax
  \loop\ifnum\@multicnt>\@ne \sp@n\repeat}
```

```
\def\sp@n{\span\omit\advance\@multicnt\m@ne}
```

`\@startpbox`
`\@endpbox`

Helper macros for 'p' columns.

`\@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)

```
\def\@startpbox#1{\vtop\bgroup \setlength\hsize{#1}\@arrayparboxrestore}
```

```
\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`

```
\let\@@startpbox=\@startpbox
\let\@@endpbox=\@endpbox
```

```
⟨/2ekernel | autoload⟩
```

File D
ltpictur.dtx

59 Picture Mode

Picture mode commands. In addition to the commands available in LaTeX2.09, This section adds the new `\qbezier` command for drawing curves.

`\qbezier` `\qbezier[`⟨*N*⟩`](`⟨*AX*,*AY*⟩`)(`⟨*BX*,*BY*⟩`)(`⟨*CX*,*CY*⟩`)` plots a quadratic Bezier curve from `(`⟨*AX*,*AY*⟩`)` to `(`⟨*CX*,*CY*⟩`)`, with `(`⟨*BX*,*BY*⟩`)` 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
 \@linefnt        = font for drawing lines
 \@circlefnt      = 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 }
                 heigth 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.

```
⟨2ekernel⟩\expandafter\let\csname ver@autopict.sty\endcsname\fmtversion
```

`\@wholewidth`
`\@halfwidth`

```
⟨*2ekernel | autoload⟩
\newdimen\@wholewidth
\newdimen\@halfwidth
```

`\unitlength`

```
\newdimen\unitlength \unitlength =1pt
```

`\@picbox`
`\@picht`

```
\newbox\@picbox
\newdimen\@picht
⟨/2ekernel | autoload⟩
```

`\picture` `#1` should be white space.

`\pictur@` `#1` should be a `(` (eating any white space before the bracket),

```
⟨*2ekernel | def⟩
\long\gdef\picture#1{\pictur@#1}
\gdef\pictur@(#1){%
  \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}
⟨/2ekernel | def⟩
⟨*autoload⟩
\def\pictur@{\@autoload{pict}}
\def\picture{\pictur@\picture}
⟨/autoload⟩
```

`\@picture`

```
⟨*2ekernel | def⟩
\gdef\@picture(#1,#2)(#3,#4){%
  \@picht#2\unitlength
  \setbox\@picbox\hb@xt@#1\unitlength\bgroup
    \hskip -#3\unitlength
    \lower #4\unitlength\hbox\bgroup
      \ignorespaces}
```

`\endpicture`

```
\gdef\endpicture{%
  \egroup\hss\egroup
    \ht\@picbox\@picht\dp\@picbox\z@
    \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).

```
\long\gdef\put(#1,#2)#3{%
  \@killglue\raise#2\unitlength
  \hb@xt@\z@{\kern#1\unitlength #3\hss}%
  \ignorespaces}
```

`\multiput` `#3` had better be a `(`.

```
\gdef\multiput(#1,#2)#3{%
  \@xdim #1\unitlength
  \@ydim #2\unitlength
   \@multiput(}
```

`\multiput`

```
\long\gdef\@multiput(#1,#2)#3#4{%
  \@killglue\@multicnt #3\relax
  \@whilenum \@multicnt >\z@\do
    {\raise\@ydim\hb@xt@\z@{\kern\@xdim #4\hss}%
     \advance\@multicnt\m@ne
     \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
  \ignorespaces}
```

`\@killglue`

```
\gdef\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}
⟨/2ekernel | def⟩
```

`\thinlines`
`\thicklines`

```
⟨*2ekernel | def⟩
\gdef\thinlines{\let\@linefnt\tenln \let\@circlefnt\tencirc
  \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
\gdef\thicklines{\let\@linefnt\tenlnw \let\@circlefnt\tencircw
  \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}
⟨/2ekernel | def⟩
⟨*autoload⟩
\def\thinlines{\pictur@\thinlines}
\def\thicklines{\pictur@\thicklines}
⟨/autoload⟩
```

`\linethickness`

```
⟨*2ekernel | def⟩
\gdef\linethickness#1{\@wholewidth #1\relax \@halfwidth .5\@wholewidth}
⟨/2ekernel | def⟩
⟨*autoload⟩
\def\linethickness{\pictur@\linethickness}
⟨/autoload⟩
```

`\ishortstack`

```
⟨*2ekernel | def⟩
\gdef\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}
```

`\@ishortstack`

```
\gdef\@shortstack[#1]{%
  \leavevmode
  \vbox\bgroup
```

```
    \baselineskip-\p@\lineskip 3\p@
    \let\mb@l\hss\let\mb@r\hss
    \expandafter\let\csname mb@#1\endcsname\relax
    \let\\\@stackcr
    \@ishortstack}
```

`\@ishortstack`

```
\gdef\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\crcr}\egroup}
```

`\@stackcr`
`\@ixstackcr`

```
\gdef\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
\gdef\@ixstackcr{\@ifnextchar[\@istackcr{\cr\ignorespaces}}
```

`\@istackcr`

```
\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
         if
 if
END

\@hvector ==
  BEGIN
    \@hline
    {\@linefnt if \@xarg < 0 then  \@getlarrow(1,0)
                             else  \@getrarrow(1,0)
                fi}
  END

\@vvector == if \@yarg < 0 \@downvector else \@upvector  fi

\@svector ==
 BEGIN
  \@sline
  \@tempcnta := |\@yarg|
    if  \@tempcnta < 5
       then  \hskip - width of \box\@linechar
             \@upordown \@clnht \hbox
                      {\@linefnt
                       if @negarg then \@getlarrow(\@xarg,\@yyarg)
                                  else \@getrarrow(\@xarg,\@yyarg)
                       fi }
       else  error: ’LATEX ERROR: Illegal \line or \vector argument.’
    fi
 END

\@getlarrow(X,Y) ==
 BEGIN
  if Y = 0
    then \@tempcnta := ’33
    else \@tempcnta := 16 * X  -  9
         \@tempcntb := 2 * Y
         if \@tempcntb > 0
           then  \@tempcnta := \@tempcnta  +  \@tempcntb
           else  \@tempcnta := \@tempcnta  -  \@tempcntb +  64
         fi
  fi
  \char\@tempcnta
 END

\@getrarrow(X,Y) ==
 BEGIN
  \@tempcntb := |Y|
  case of \@tempcntb
    0 : \@tempcnta := ’55
    1 : if X < 3
          then \@tempcnta :=  24*X - 6
          else if X = 3
                 then \@tempcnta := 49
                 else \@tempcnta := 58  fi
        fi
    2 : if X < 3
          then \@tempcnta :=  24*X - 3
          else \@tempcnta := 51      % X must = 3
        fi
```

```
      3 : \@tempcnta := 16*X - 2
      4 : \@tempcnta := 16*X + 7
    endcase
    if Y < 0
      then \@tempcnta := \@tempcnta + 64
    fi
    \char\@tempcnta
  END
```

`\if@negarg`

```
\newif\if@negarg
```

`\line`

```
\gdef\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@linelen #3\unitlength
  \ifdim\@linelen<\z@\@badlinearg\else
    \ifnum\@xarg =\z@ \@vline
      \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
    \fi
  \fi}
```

`\@sline`

```
\gdef\@sline{%
  \ifnum\@xarg<\z@ \@negargtrue \@xarg -\@xarg \@yyarg -\@yarg
  \else \@negargfalse \@yyarg \@yarg \fi
\ifnum \@yyarg >\z@ \@tempcnta\@yyarg \else \@tempcnta -\@yyarg \fi
\ifnum\@tempcnta>6 \@badlinearg\@tempcnta\z@ \fi
\ifnum\@xarg>6 \@badlinearg\@xarg \@ne \fi
\setbox\@linechar\hbox{\@linefnt\@getlinechar(\@xarg,\@yyarg)}%
```

If we have something like `\line(5,5){30}` the `\@linechar` will not contain a char and later on we will end in an infinite loop. So we check the width of the box and put in something as an emergency fix if necessary.

```
\ifdim\wd\@linechar=\z@
   \setbox\@linechar\hbox{.}%
   \@badlinearg
\fi
\ifnum \@yarg >\z@ \let\@upordown\raise \@clnht\z@
   \else\let\@upordown\lower \@clnht \ht\@linechar\fi
\@clnwd \wd\@linechar
\if@negarg
  \hskip -\wd\@linechar \def\reserved@a{\hskip -2\wd\@linechar}%
\else
     \let\reserved@a\relax
\fi
\@whiledim \@clnwd <\@linelen \do
  {\@upordown\@clnht\copy\@linechar
   \reserved@a
   \advance\@clnht \ht\@linechar
   \advance\@clnwd \wd\@linechar}%
\advance\@clnht -\ht\@linechar
\advance\@clnwd -\wd\@linechar
\@tempdima\@linelen\advance\@tempdima -\@clnwd
\@tempdimb\@tempdima\advance\@tempdimb -\wd\@linechar
\if@negarg \hskip -\@tempdimb \else \hskip \@tempdimb \fi
\multiply\@tempdima \@m
\@tempcnta \@tempdima
\@tempdima \wd\@linechar \divide\@tempcnta \@tempdima
\@tempdima \ht\@linechar \multiply\@tempdima \@tempcnta
\divide\@tempdima \@m
\advance\@clnht \@tempdima
```

```
\ifdim \@linelen <\wd\@linechar
   \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).

```
   \ifdim \@linelen = \z@
   \else
     \@picture@warn
   \fi
   \else\@upordown\@clnht\copy\@linechar\fi}
```

`\@hline`

```
\gdef\@hline{\ifnum \@xarg <\z@ \hskip -\@linelen \fi
\vrule \@height \@halfwidth \@depth \@halfwidth \@width \@linelen
\ifnum \@xarg <\z@ \hskip -\@linelen \fi}
```

`\getlinechar`

```
\gdef\@getlinechar(#1,#2){\@tempcnta#1\relax\multiply\@tempcnta 8%
  \advance\@tempcnta -9\ifnum #2>\z@ \advance\@tempcnta #2\relax\else
  \advance\@tempcnta -#2\relax\advance\@tempcnta 64 \fi
  \char\@tempcnta}
```

`\vector`

```
\gdef\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
  \ifnum\@tempcnta<5\relax
  \@linelen #3\unitlength
  \ifdim\@linelen<\z@\@badlinearg\else
    \ifnum\@xarg =\z@ \@vvector
      \else \ifnum\@yarg =\z@ \@hvector \else \@svector\fi
    \fi
  \fi
  \else\@badlinearg\fi}
```

`\@hvector`

```
\gdef\@hvector{\@hline\hb@xt@\z@{\@linefnt
 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
    \hss\@getrarrow(1,0)\fi}}
```

`\@vvector`

```
\gdef\@vvector{\ifnum \@yarg <\z@ \@downvector \else \@upvector \fi}
```

`\@svector`

```
\gdef\@svector{\@sline
  \@tempcnta\@yarg \ifnum\@tempcnta <\z@ \@tempcnta -\@tempcnta\fi
  \ifnum\@tempcnta <5%
    \hskip -\wd\@linechar
    \@upordown\@clnht \hbox{\@linefnt  \if@negarg
    \@getlarrow(\@xarg,\@yyarg)\else \@getrarrow(\@xarg,\@yyarg)\fi}%
  \else\@badlinearg\fi}
```

`\@getlarrow`

```
\gdef\@getlarrow(#1,#2){\ifnum #2=\z@ \@tempcnta 27 % '33
  \else
  \@tempcnta #1\relax\multiply\@tempcnta \sixt@@n
  \advance\@tempcnta -9 \@tempcntb #2\relax\multiply\@tempcntb \tw@
  \ifnum \@tempcntb >\z@ \advance\@tempcnta \@tempcntb
  \else\advance\@tempcnta -\@tempcntb\advance\@tempcnta 64
  \fi\fi\char\@tempcnta}
```

\@getrarrow

```
\gdef\@getrarrow(#1,#2){\@tempcntb #2\relax
\ifnum\@tempcntb <\z@ \@tempcntb -\@tempcntb\relax\fi
\ifcase \@tempcntb\relax \@tempcnta 45  % ’55
\or
\ifnum #1<\thr@@ \@tempcnta #1\relax\multiply\@tempcnta
24\advance\@tempcnta -6 \else \ifnum #1=\thr@@ \@tempcnta 49
\else\@tempcnta 58 \fi\fi\or
\ifnum #1<\thr@@ \@tempcnta=#1\relax\multiply\@tempcnta
24\advance\@tempcnta -\thr@@ \else \@tempcnta 51 \fi\or
\@tempcnta #1\relax\multiply\@tempcnta
\sixt@@n \advance\@tempcnta -\tw@ \else
\@tempcnta #1\relax\multiply\@tempcnta
\sixt@@n \advance\@tempcnta 7 \fi\ifnum #2<\z@ \advance\@tempcnta 64 \fi
\char\@tempcnta}
```

\@vline

```
\gdef\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}
```

\@upline

```
\gdef\@upline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \@linelen \@depth \z@\hss}}
```

\@downline

```
\gdef\@downline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \z@ \@depth \@linelen \hss}}
```

\@upvector

```
\gdef\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}%  % ’66
  \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}
```

\@downvector

```
\gdef\@downvector{\@downline\lower \@linelen
     \hb@xt@\z@{\@linefnt\char 63  % ’77
     \hss}}
```

```
\dashbox{D}(X,Y) ==
 BEGIN
 leave vertical mode
 \hb@xt@ 0pt {
      \baselineskip := 0pt
      \lineskip     := 0pt
 %% HORIZONTAL DASHES
      \@dashdim := X * \unitlength
      \@dashcnt := \@dashdim + 200 % to prevent roundoff error
      \@dashdim := D * \unitlength
      \@dashcnt := \@dashcnt / \@dashdim
      if \@dashcnt is odd
        then \@dashdim := 0pt
             \@dashcnt := (\@dashcnt + 1) / 2
        else \@dashdim := \@dashdim / 2
             \@dashcnt := \@dashcnt / 2 - 1
             \box\@dashbox   := \hbox{\vrule height \@halfwidth
                                depth \@halfwidth width \@dashdim}
             \put(0,0){\copy\@dashbox}
             \put(0,Y){\copy\@dashbox}
             \put(X,0){\hskip -\@dashdim\copy\@dashbox}
             \put(X,Y){\hskip -\@dashdim\box\@dashbox}
             \@dashdim := 3 * \@dashdim
      fi
      \box\@dashbox := \hbox{\vrule height \@halfwidth
                        depth \@halfwidth width D * \unitlength
                        \hskip D * \unitlength}
      \@tempcnta := 0
      \put(0,0){\hskip \@dashdim
               while \@tempcnta < \@dascnt
                 do \copy\@dashbox
                    \@tempcnta := \@tempcnta + 1
                 od
              }
      \@tempcnta := 0
      put(0,Y){\hskip \@dashdim
               while \@tempcnta < \@dascnt
                 do \copy\@dashbox
                    \@tempcnta := \@tempcnta + 1
                 od
              }

%% vertical dashes
      \@dashdim := Y * \unitlength
      \@dashcnt := \@dashdim + 200 % to prevent roundoff error
      \@dashdim := D * \unitlength
      \@dashcnt := \@dashcnt / \@dashdim
      if \@dashcnt is odd
        then \@dashdim := 0pt
             \@dashcnt := (\@dashcnt + 1) / 2
        else \@dashdim := \@dashdim / 2
             \@dashcnt := \@dashcnt / 2 - 1
             \box\@dashbox   := \hbox{\hskip -\@halfwidth
                                      \vrule width \@wholewidth
                                               height \@dashdim  }
             \put(0,0){\copy\@dashbox}
             \put(X,0){\copy\@dashbox}
             \put(0,Y){\lower\@dashdim\copy\@dashbox}
             \put(X,Y){\lower\@dashdim\copy\@dashbox}
             \@dashdim := 3 * \@dashdim
      fi
      \box\@dashbox := \hbox{\vrule width \@wholewidth
                                height D * \unitlength       }
      \@tempcnta := 0
      put(0,0){\hskip -\halfwidth
               \vbox{while \@tempcnta < \@dashcnt
                      do \vskip D*\unitlength
                         \copy\@dashbox
                         \@tempcnta := \@tempcnta + 1
                      od
                     \vskip \@dashdim
                    } }
      \@tempcnta := 0
      put(X,0){\hskip -\halfwidth
               \vbox{while \@tempcnta < \@dashcnt
                      do \vskip D*\unitlength
                         \copy\@dashbox
                         \@tempcnta := \@tempcnta + 1
                      od
                     \vskip \@dashdim
                    }
              }
   }     % END DASHES

 \@imakepicbox(X,Y)
END
```

\dashbox

```
\gdef\dashbox#1(#2,#3){\leavevmode\hb@xt@\z@{\baselineskip \z@skip
\lineskip \z@skip
\@dashdim #2\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
\ifodd\@dashcnt\@dashdim \z@
\advance\@dashcnt \@ne \divide\@dashcnt \tw@
\else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
\advance\@dashcnt \m@ne
\setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
\@width \@dashdim}\put(0,0){\copy\@dashbox}%
\put(0,#3){\copy\@dashbox}%
\put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
\put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
\multiply\@dashdim \thr@@
\fi
\setbox\@dashbox \hbox{\vrule \@height \@halfwidth \@depth \@halfwidth
\@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
\put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
\do{\copy\@dashbox\advance\@tempcnta \@ne }}\@tempcnta\z@
\put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
\do{\copy\@dashbox\advance\@tempcnta \@ne }}%
\@dashdim #3\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
\ifodd\@dashcnt \@dashdim \z@
\advance\@dashcnt \@ne \divide\@dashcnt \tw@
\else
\divide\@dashdim \tw@ \divide\@dashcnt \tw@
\advance\@dashcnt \m@ne
\setbox\@dashbox\hbox{\hskip -\@halfwidth
\vrule \@width \@wholewidth
\@height \@dashdim}\put(0,0){\copy\@dashbox}%
\put(#2,0){\copy\@dashbox}%
\put(0,#3){\lower\@dashdim\copy\@dashbox}%
\put(#2,#3){\lower\@dashdim\copy\@dashbox}%
\multiply\@dashdim \thr@@
\fi
\setbox\@dashbox\hbox{\vrule \@width \@wholewidth
\@height #1\unitlength}\@tempcnta\z@
\put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
\do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}\@tempcnta\z@
\put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
\do{\vskip #1\unitlength\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}}\@makepicbox(#2,#3)}
```

CIRCLES AND OVALS

USER COMMANDS:

`\circle{D}` : Produces the circle with the diameter as close as possible to D * `\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 * `\unitlength` and height Y * `\unitlength`. The reference point is the center.

`\oval(X,Y)[POS]` : Save as `\oval(X,Y)` except it draws only the half or quadrant of the oval indicated by POS. E.G., `\oval(X,Y)[t]` draws just the top half and `\oval(X,Y)[br]` draws just the bottom right quadrant. In all cases, the reference point is the same as the unqualified `\oval(X,Y)` command.

`\@ovvert {DELTA1} {DELTA2}` : Makes a vbox containing either the left side or the right side of the oval being constructed. The baseline will coincide with the outside bottom edge of the oval; the left side of the box will coincide with the left edge of the vertical rule. The width of the box will be `\@tempdima`.
DELTA1 and DELTA2 are added to the character number in
`\@tempcnta`
to get the characters for the top and bottom quarter circle pieces.

`\@ovhorz` : Makes an hbox containing the straight rule for either the top or the bottom of the oval being constructed. The baseline will coincide with bottom edge of the rule; the left side of the box will coincide with the left side of the oval. The width of the box will be `\@ovxx`.

`\@getcirc {DIAM}` : Sets `\@tempcnta` to the character number of the top-right quarter circle with the largest diameter less than or equal to DIAM.
Sets `\@tempboxa` to an hbox containing that character.
Sets `\@tempdima` to `\wd \@tempboxa`, which is the distance from the circle's left outside edge to its right inside edge.
(These characters are like those described in the TeXbook, pp. 389-90.)

```
\@getcirc {DIAM} ==
  BEGIN
    \@tempcnta       := integer coercion of (DIAM + 2pt)
                                          + 2pt added 1 Nov 88
    \@tempcnta       := \@tempcnta / integer coercion of 4pt
    if \@tempcnta > 10
      then \@tempcnta := 10 fi
    if \@tempcnta > 0
      then \@tempcnta := \@tempcnta-1
      else LaTeX Warning: Oval too small.
    fi
    \@tempcnta       := 4 * \@tempcnta
    \@tempboxa       := \hbox{\@circlefnt \char \@tempcnta}
    \@tempdima       := \wd \@tempboxa
  END

\@put{X}{Y}{OBJ} ==
   BEGIN
     \raise Y \hb@xt@ 0pt{\hskip X OBJ \hss}
   END

 \@oval(X,Y)[POS] ==
   BEGIN
     \begingroup
       \boxmaxdepth := \maxdimen
       @ovt := @ovb := @ovl := @ovr := true
       for all E in POS
         do  @ovE := false od
       \@ovxx      := X * \unitlength
       \@ovyy      := Y * \unitlength
       \@tempdimb := min(\@ovxx,\@ovyy)
       \@getcirc{\@tempdimb-2pt}   %% "-2pt" added 7 Dec 89
       \@ovro     := \ht \@tempboxa
       \@ovri     := \dp \@tempboxa
       \@ovdx     := \@ovxx - \@tempdima
       \@ovdx     := \@ovdx/2
       \@ovdy     := \@ovyy - \@tempdima
       \@ovdy     := \@ovyy/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@ovb` `\if@ovl` `\if@ovr`

If producing the Top Bottom Left or Right of an oval.

```
\newif\if@ovt
\newif\if@ovb
\newif\if@ovl
\newif\if@ovr
```

```
⟨/2ekernel | def⟩
⟨*2ekernel | autoload⟩
```

`\@ovxx` `\@ovyy` `\@ovdx` `\@ovdy` `\@ovro` `\@ovri`

```
\newdimen\@ovxx
\newdimen\@ovyy
\newdimen\@ovdx
\newdimen\@ovdy
\newdimen\@ovro
\newdimen\@ovri
```

```
⟨/2ekernel | autoload⟩
```

`\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.

```
⟨*2ekernel | def⟩
```

`\@getcirc`

```
\gdef\@getcirc#1{\@tempdima #1\relax \advance\@tempdima 2\p@
  \@tempcnta\@tempdima
  \@tempdima 4\p@ \divide\@tempcnta\@tempdima
  \ifnum \@tempcnta >10\relax
      \@picture@warn
      \@tempcnta 10\relax
  \fi
  \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne
```

Warn if requirements for oval or circle can't be met.

```
    \else \@picture@warn \fi
  \multiply\@tempcnta 4\relax
  \setbox \@tempboxa \hbox{\@circlefnt
  \char \@tempcnta}\@tempdima \wd \@tempboxa}
```

`\@picture@warn`

Generic warning for lines, vectors (used in `\@sline`) and oval or circle (used un `\@getcirc`) are not available at right size.

```
\def\@picture@warn{\@latex@warning{%
    \string\oval, \string\circle, or \string\line\space
    size unavailable}}
```

`\@put`

```
\gdef\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}
```

`\oval`

```
\gdef\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2)[]}}
```

`\@oval`

```
\gdef\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
  \@tfor\reserved@a :=#3\do{\csname @ov\reserved@a false\endcsname}%
  \@ovxx
  #1\unitlength \@ovyy #2\unitlength
  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
  \advance \@tempdimb -2\p@
  \@getcirc \@tempdimb
  \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
  \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
  \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
  \@circlefnt \setbox\@tempboxa
  \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
  \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
  \if@ovt \@ovhorz \kern -\@ovxx \fi
  \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
  \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
  \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
  \endgroup}
```

`\@ovvert`

```
\gdef\@ovvert#1#2{\vbox to\@ovyy{%
    \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
      \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
    \else \kern \@ovri \kern \@ovdy \fi
    \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
    \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
      \hbox{\char \@tempcntb}%
    \else \kern \@ovdy \kern \@ovro \fi}}
```

`\@ovhorz`

```
\gdef\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
    \if@ovr \else \kern \@ovdx \fi
    \leaders \hrule \@height \@wholewidth \hfil
    \if@ovl \else \kern \@ovdx \fi
    \kern \@ovri}}
```

`\circle`

```
\gdef\circle{\@inmatherr\circle\@ifstar\@dot\@circle}
```

`\@circle`

```
\gdef\@circle#1{%
  \begingroup \boxmaxdepth \maxdimen \@tempdimb #1\unitlength
   \ifdim \@tempdimb >15.5\p@ \@getcirc\@tempdimb
      \@ovro\ht\@tempboxa
     \setbox\@tempboxa\hbox{\@circlefnt
      \advance\@tempcnta\tw@ \char \@tempcnta
      \advance\@tempcnta\m@ne \char \@tempcnta \kern -2\@tempdima
      \advance\@tempcnta\tw@
      \raise \@tempdima \hbox{\char\@tempcnta}\raise \@tempdima
        \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
      \@put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
   \else  \@circ\@tempdimb{96}\fi\endgroup}
```

`\@dot` Internal form of `\circle*`.

```
\gdef\@dot#1{\@tempdimb #1\unitlength \@circ\@tempdimb{112}}
```

`\@circ`

```
\gdef\@circ#1#2{\@tempdima #1\relax \advance\@tempdima .5\p@
   \@tempcnta\@tempdima \@tempdima \p@
```

```
  \divide\@tempcnta\@tempdima
  \ifnum\@tempcnta >15\relax \@tempcnta 15\relax \fi
  \ifnum \@tempcnta >\z@ \advance\@tempcnta\m@ne\fi
  \advance\@tempcnta #2\relax
  \@circlefnt \char\@tempcnta}
⟨/2ekernel | def⟩
⟨*2ekernel | autoload⟩
```

`\@xarg` `\@yarg` `\@yyarg` Counters used for manipulating the 'slope' arguments.

```
\newcount\@xarg
\newcount\@yarg
\newcount\@yyarg
```

`\@multicnt` Counter used in `\multiput`, and also `\multicolumn`.

```
\newcount\@multicnt
```

`\@xdim` `\yxdim` Length registers.

```
\newdimen\@xdim
\newdimen\@ydim
```

`\@linechar` Box for holding a line segment character, for sloping lines.

```
\newbox\@linechar
```

`\@linelen` Length of the line currently being built.

```
\newdimen\@linelen
```

`\@clnwd` `\@clnht` Height and width of current line segment.

```
\newdimen\@clnwd
\newdimen\@clnht
```

`\@dashdim` `\@dashbox` `\@dashcnt` `\dashbox` internal registers.

```
\newdimen\@dashdim
\newbox\@dashbox
\newcount\@dashcnt
```

Initialization: "`\thinlines`"

```
\let\@linefnt\tenln
\let\@circlefnt\tencirc
\@wholewidth\fontdimen8\tenln
\@halfwidth .5\@wholewidth
⟨/2ekernel | autoload⟩
```

59.1 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.

```
⟨*2ekernel | def⟩
⟨def⟩\ifx\qbeziermax\@undefined
\gdef\qbeziermax{500}
⟨def⟩\fi
```

In the code below, to save registers `\@a` … are not used. Instead other registers are reused.

```
\newcounter{@sc} -> \c@multicnt
\newcounter{@scp} -> \@tempcnta
\newdimen\@xa ->  \@ovxx
\newdimen\@xb ->  \@ovdx
\newdimen\@ya ->  \@ovyy
\newdimen\@yb ->  \@ovdy
\newsavebox{\@pictdot} -> \@tempboxa
```

`\qbezier` Main user-level command to plot quadratic bezier curves. `#2` should be `(`.

```
\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 `(`.

```
\gdef\bezier#1)#2(#3)#4({\@bezier#1)(#3)(}
```

`\@bezier`

```
\gdef\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
 \ifnum #1=\z@
     \@ovxx #4\unitlength
       \advance\@ovxx -#2\unitlength
       \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
     \@ovdx #6\unitlength
       \advance\@ovdx -#4\unitlength
       \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
       \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
     \@ovyy #5\unitlength
       \advance\@ovyy -#3\unitlength
```

```
      \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
    \@ovdy #7\unitlength
      \advance\@ovdy -#5\unitlength
      \ifdim \@ovdy<\z@  \@ovdy -\@ovdy \fi
      \ifdim \@ovyy<\@ovdy \@ovyy  \@ovdy \fi
    \@multicnt
       \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
    \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
    \ifnum \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax \fi
\else \@multicnt#1\relax \fi
\@tempcnta\@multicnt \advance\@tempcnta\@ne
\@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
    \multiply\@ovdx \tw@
\@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
    \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
\@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
     \multiply\@ovdy \tw@
\@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
    \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt

\setbox\@tempboxa\hbox{%
          \hskip -\@halfwidth
          \vrule \@height\@halfwidth
                 \@depth \@halfwidth
                 \@width \@wholewidth}%
 \put(#2,#3){%
   \count@\z@
   \@whilenum{\count@<\@tempcnta}\do
      {\@xdim\count@\@ovxx
         \advance\@xdim\@ovdx
         \divide\@xdim\@multicnt
         \multiply\@xdim\count@
       \@ydim\count@\@ovyy
          \advance\@ydim\@ovdy
          \divide\@ydim\@multicnt
          \multiply\@ydim\count@
       \raise \@ydim
          \hb@xt@\z@{\kern\@xdim
                     \unhcopy\@tempboxa\hss}%
       \advance\count@\@ne}}}
⟨/2ekernel | def⟩
```

File E

ltthm.dtx

60 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 ⟨*name*⟩ to be just as one would expect a theorem environment to be, except that it prints ⟨*text*⟩ instead of “Theorem”.

If ⟨*oldname*⟩ is given, then environments ⟨*name*⟩ and ⟨*oldname*⟩ use the same counter, so using a ⟨*name*⟩ environment advances the number of the next ⟨*name*⟩ environment, and vice-versa.

If ⟨*counter*⟩ is given, then environment ⟨*name*⟩ is numbered within ⟨*counter*⟩.

E.g., if ⟨*counter*⟩ = `subsection`, then the first ⟨*name*⟩ in subsection 7.2 is numbered ⟨*text*⟩ 7.2.1.

The way ⟨*name*⟩ environments are numbered can be changed by redefining `\the`⟨*name*⟩.

```
  DOCUMENT  STYLE  PARAMETERS

  \@thmcounter{COUNTER} : A  command  such  that
                \edef\theCOUNTER{\@thmcounter{COUNTER}}
     defines \theCOUNTER to produce a number for a theorem environment.
     The  default  is:
            BEGIN \noexpand\arabic{COUNTER} END

  \@thmcountersep : A  separator  placed  between  a  theorem  number  and
         the  number  of  the  counter  within  which  it  is  numbered.
         E.g.,  to  make  the  third  theorem  of  section  7.2  be  numbered
         7.2-3, \@thmcountersep should be \def'ed to '-'.  Its
         default  is  '.'.

  \@begintheorem{NAME}{NUMBER} : A  command  that  begins  a
theorem
         environment  for  a  'theorem'  named  'NAME  NUMBER'  –
         e.g., \@begintheorem{Lemma}{3.7} starts  Lemma  3.7.

  \@opargbegintheorem{NAME}{NUMBER}{OPARG} :
         A  command  that  begins  a  theorem
         environment  for  a  'theorem'  named  'NAME  NUMBER'  with
optional
         argument  OPARG  –  e.g., \@begintheorem{Lemma}{3.7}{Jones}
         starts  'Lemma  3.7  (Jones):'.

  \@endtheorem : A  command  that  ends  a  theorem  environment.

 \newtheorem{NAME}{TEXT}[COUNTER]  ==
   BEGIN
     if \NAME is definable
       then \@definecounter{NAME}
            if COUNTER present
              then \@newctr{NAME}[COUNTER] fi
                   \theNAME ==  BEGIN \theCOUNTER \@thmcountersep
                                      eval\@thmcounter{NAME}
END
```

```
                else \theNAME ==  BEGIN eval\@thmcounter{NAME} END
              \NAME == \@thm{NAME}{TEXT}
              \endNAME == \@endtheorem
        else  error
      fi
    END

  \newtheorem{NAME}[OLDNAME]{TEXT}==
    BEGIN
      if counter OLDNAME nonexistant
        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…

```
⟨*2ekernel⟩
\def\newtheorem#1{%
  \@ifnextchar[{\@othm{#1}}{\@nthm{#1}}}
```

`\@nthm`

```
\def\@nthm#1#2{%
  \@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`)

```
\def\@xnthm#1#2[#3]{%
  \expandafter\@ifdefinable\csname #1\endcsname
    {\@definecounter{#1}\@newctr{#1}[#3]%
     \expandafter\xdef\csname the#1\endcsname{%
       \expandafter\noexpand\csname the#3\endcsname \@thmcountersep
          \@thmcounter{#1}}%
     \global\@namedef{#1}{\@thm{#1}{#2}}%
     \global\@namedef{end#1}{\@endtheorem}}}
```

`\@ynthm`

```
\def\@ynthm#1#2{%
  \expandafter\@ifdefinable\csname #1\endcsname
    {\@definecounter{#1}%
     \expandafter\xdef\csname the#1\endcsname{\@thmcounter{#1}}%
     \global\@namedef{#1}{\@thm{#1}{#2}}%
     \global\@namedef{end#1}{\@endtheorem}}}
```

`\@othm`

```
\def\@othm#1[#2]#3{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}%
    {\expandafter\@ifdefinable\csname #1\endcsname
    {\global\@namedef{the#1}{\@nameuse{the#2}}%
  \global\@namedef{#1}{\@thm{#2}{#3}}%
  \global\@namedef{end#1}{\@endtheorem}}}}
```

`\@thm`

```
\def\@thm#1#2{%
  \refstepcounter{#1}%
  \@ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
```

`\@xthm`
`\@ythm`

```
\def\@xthm#1#2{%
  \@begintheorem{#2}{\csname the#1\endcsname}\ignorespaces}
\def\@ythm#1#2[#3]{%
  \@opargbegintheorem{#2}{\csname the#1\endcsname}{#3}\ignorespaces}
```

Default values

`\@thmcounter`
`\@thmcountersep`

```
\def\@thmcounter#1{\noexpand\arabic{#1}}
\def\@thmcountersep{.}
```

`\@begintheorem`
`\@opargbegintheorem`
`\@endtheorem`

Providing theorem defaults.

```
\def\@begintheorem#1#2{\trivlist
   \item[\hskip \labelsep{\bfseries #1\ #2}]\itshape}
\def\@opargbegintheorem#1#2#3{\trivlist
       \item[\hskip \labelsep{\bfseries #1\ #2\ (#3)}]\itshape}
\def\@endtheorem{\endtrivlist}
⟨/2ekernel⟩
```

File F
ltsect.dtx

61 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 LaTeX 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.

```
⟨*2ekernel⟩
\message{title,}
```

61.1 The Title

`\title` `\author` `\date` `\thanks` `\and`

The user defines the title and author by the declarations `\title{`⟨*name*⟩`}`, `\author{`⟨*name*⟩`}`

Similarly the date is declared with `\date{`⟨*date*⟩`}`.

Inside these, the `\thanks{`⟨*footnote text*⟩`}` command may be used to make acknowledgements, notice of address, etc. in a footnote. If there are multiple authors, they have to be separated with the `\and` command.

`\maketitle`

And finally, the `\maketitle` command produces the actual title, using the information previously saved with the other commands.

`\title` `\@title`

`\title` for use in `\maketitle`. If not given `\maketitle` will produce an error message.

```
\def\title#1{\gdef\@title{#1}}
\def\@title{\@latex@error{No \noexpand\title given}\@ehc}
```

`\author` `\@author`

`\author` for use in `\maketitle`. If not given `\maketitle` will produce a warning message.

```
\def\author#1{\gdef\@author{#1}}
\def\@author{\@latex@warning@no@line{No \noexpand\author given}}
```

`\date` `\@date`

`\date` for use in `\maketitle`. If not given `\maketitle` will produce `\today` as the default.

```
\def\date#1{\gdef\@date{#1}}
\gdef\@date{\today}
```

`\thanks`

```
\def\thanks#1{\footnotemark
    \protected@xdef\@thanks{\@thanks
        \protect\footnotetext[\the\c@footnote]{#1}}%
}
```

`\@thanks`

```
\let\@thanks\@empty
```

`\and`

```
\def\and{%                  % \begin{tabular}
  \end{tabular}%
  \hskip 1em \@plus.17fil%
  \begin{tabular}[t]{c}}%   % \end{tabular}
```

```
\message{sectioning,}
```

61.2 Sectioning

`\@secpenalty`

```
\newcount\@secpenalty
\@secpenalty = -300
```

`\if@noskipsec`
`\@noskipsectrue`

Way back in 1991 (08/26) FMi & RmS set the `\@noskipsec` switch to true for the preamble and to false in `\document`. This was done to trap lists and related text in the preamble but it does not catch everything.

```
\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}
```

```
             \addvspace{\@tempskipa}
      FI
      IF * next
         THEN \@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
         ELSE \@dblarg{\@sect
                  {NAME}{LEVEL}{INDENT}
                  {BEFORESKIP}{AFTERSKIP}{STYLE}}
      FI
 END
```

```
\def\@startsection#1#2#3#4#5#6{%
  \if@noskipsec \leavevmode \fi
  \par
  \@tempskipa #4\relax
  \@afterindenttrue
  \ifdim \@tempskipa <\z@
    \@tempskipa -\@tempskipa \@afterindentfalse
  \fi
  \if@nobreak
    \everypar{}%
  \else
    \addpenalty\@secpenalty\addvspace\@tempskipa
  \fi
  \@ifstar
    {\@ssect{#3}{#4}{#5}{#6}}%
    {\@dblarg{\@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}
```

`\@sect` Pseudocode for the `\@sect` command

```
\@sect{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}[ARG1]{ARG2}
           ==
   BEGIN
    IF LEVEL > \c@secnumdepth
       THEN \@svsec :=L null
       ELSE \refstepcounter{NAME}
            \@svsec :=L BEGIN \@seccntformat{#1}\relax END
    FI
    IF AFTERSKIP > 0
       THEN \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
```

```
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
    \let\@svsec\@empty
  \else
    \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.

```
    \protected@edef\@svsec{\@seccntformat{#1}\relax}%
  \fi
  \@tempskipa #5\relax
  \ifdim \@tempskipa>\z@
    \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`.

```
      #6{%
        \@hangfrom{\hskip #3\relax\@svsec}%
          \interlinepenalty \@M #8\@@par}%
    \endgroup
    \csname #1mark\endcsname{#7}%
    \addcontentsline{toc}{#1}{%
      \ifnum #2>\c@secnumdepth \else
        \protect\numberline{\csname the#1\endcsname}%
      \fi
      #7}%
  \else
```

`\relax` added 2 May 90

```
    \def\@svsechd{%
      #6{\hskip #3\relax
      \@svsec #8}%
      \csname #1mark\endcsname{#7}%
      \addcontentsline{toc}{#1}{%
        \ifnum #2>\c@secnumdepth \else
          \protect\numberline{\csname the#1\endcsname}%
        \fi
        #7}}%
  \fi
  \@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

END
```

```
\def\@xsect#1{%
  \@tempskipa #1\relax
  \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.

```
    \par \nobreak
    \vskip \@tempskipa
    \@afterheading
  \else
```

```
    \@nobreakfalse
    \global\@noskipsectrue
    \everypar{%
      \if@noskipsec
        \global\@noskipsecfalse
       {\setbox\z@\lastbox}%
        \clubpenalty\@M
        \begingroup \@svsechd \endgroup
        \unskip
        \@tempskipa #1\relax
        \hskip -\@tempskipa
      \else
        \clubpenalty \@clubpenalty
        \everypar{}%
      \fi}%
  \fi
  \ignorespaces}
```

`\@seccntformat` This command formats the section number including the space following it.

```
\def\@seccntformat#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`

```
\def\@ssect#1#2#3#4#5{%
  \@tempskipa #3\relax
  \ifdim \@tempskipa>\z@
    \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`.

```
      #4{%
        \@hangfrom{\hskip #1}%
          \interlinepenalty \@M #5\@@par}%
    \endgroup
  \else

    \def\@svsechd{#4{\hskip #1\relax #5}}%
  \fi
  \@xsect{#3}}
```

`\if@afterindent`
`\@afterindenttrue`

```
\newif\if@afterindent \@afterindenttrue
```

`\@afterheading` This hook is used in setting up custom-built headings in classes.dtx.

```
\def\@afterheading{%
  \@nobreaktrue
  \everypar{%
    \if@nobreak
      \@nobreakfalse
      \clubpenalty \@M
      \if@afterindent \else
        {\setbox\z@\lastbox}%
      \fi
    \else
      \clubpenalty \@clubpenalty
      \everypar{}%
    \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.

```
\def\@hangfrom#1{\setbox\@tempboxa\hbox{{#1}}%
      \hangindent \wd\@tempboxa\noindent\box\@tempboxa}
```

`\c@secnumdepth`
`\c@tocdepth`

```
\newcount\c@secnumdepth
\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*{...}`

```
\def\secdef#1#2{\@ifstar{#2}{\@dblarg{#1}}}
```

61.2.1 Initializations

`\sectionmark` `\subsectionmark` `\subsubsectionmark` `\paragraphmark` `\subparagraphmark`

```
\let\sectionmark\@gobble
\let\subsectionmark\@gobble
\let\subsubsectionmark\@gobble
\let\paragraphmark\@gobble
\let\subparagraphmark\@gobble
```

```
\message{contents,}
```

61.3 Table of Contents etc.

61.3.1 Convention

`\tf@`⟨*foo*⟩ = file number for output for table foo. The file is opened only if `@filesw` = `true`.

61.3.2 Commands

A `\l@`⟨*type*⟩`{`⟨*entry*⟩`}{`⟨*page*⟩`}` Macro needs to 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
```

```
\def\@starttoc#1{%
\begingroup
  \makeatletter
  \@input{\jobname.#1}%
  \if@filesw
    \expandafter\newwrite\csname tf@#1\endcsname
    \immediate\openout \csname tf@#1\endcsname \jobname.#1\relax
  \fi
  \@nobreakfalse
\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 aplication of `\addtocontents`. Note that `\thepage` is not expandable during `\protected@write` therefore one gets the page number at the time of the `\shipout`.

```
\def\addcontentsline#1#2#3{%
  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}
```

`\addtocontents`

The `\addtocontents{`⟨*table*⟩`}{`⟨*text*⟩`}` command adds ⟨*text*⟩ to the `.`⟨*table*⟩ file, with no page number.

```
\long\def\addtocontents#1#2{%
  \protected@write\@auxout
      {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
      {\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.

```
\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`

```
\def\@dottedtocline#1#2#3#4#5{%
  \ifnum #1>\c@tocdepth \else
    \vskip \z@ \@plus.2\p@
    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
     \parindent #2\relax\@afterindenttrue
     \interlinepenalty\@M
     \leavevmode
     \@tempdima #3\relax

     \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
     {#4}\nobreak
     \leaders\hbox{$\m@th
```

If a document uses fonts other than computer modern, the use of a dot from math can be very disturbing despite the fact that this might be the only place in a document that then uses computer modern. Therefore we surround the dot with an `\hbox` to escape to the surrounding text font.

```
        \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
        mu$}\hfill
     \nobreak
     \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
     \par}%
  \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.)

```
\def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
⟨/2ekernel⟩
```

File G
ltfloat.dtx

62 Floats

The different types of floats are identified by a ⟨*type*⟩ 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 ⟨*type*⟩ has associated a positive ⟨*type number*⟩, 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 ⟨*placement specifier*⟩, 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.

62.1 Floating Environments

```
⟨*2ekernel⟩
\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
   \@captype ==L TYPE
   \@dblflset
   \@fps     ==L PLACEMENT
   \@onelevel@sanitize \@fps
   add default PLACEMENT if at most ! in PLACEMENT == \@fpsadddefault
   if inner
     then LaTeX Error: 'Not in outer paragraph mode.'
          \@floatpenalty := 0
     else if \@freelist nonempty
            then \@currbox  :=L head of \@freelist
                 \@freelist :=G tail of \@freelist
                 \count\@currbox :=G 32*\ftype@TYPE +
                                          bits determined by PLACEMENT
            else \@floatpenalty := 0
                 LaTeX Error: 'Too many unprocessed floats'
          fi
   fi
   \@currbox :=G   \color@vbox
                     \normalcolor
                       \vbox{
                        %% 15 Dec 87 -
                        %% removed \boxmaxdepth :=L 0pt
                        %% that made box 0 depth because it screwed
                        %% things up. Instead, added \vskip0pt at end
                              \hsize = \columnwidth
                              \@parboxrestore
                              \@floatboxreset
 END

\caption ==
  BEGIN
   \refstepcounter{\@captype}
   \@dblarg{\@caption{\@captype}}
  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`

```
\newcount\@floatpenalty
```

`\caption` This is set to be an error message outside a float since no captype is defined there; this may need to be changed by some classes.

```
\def\caption{%
   \ifx\@captype\@undefined
     \@latex@error{\noexpand\caption outside float}\@ehd
     \expandafter\@gobble
   \else
     \refstepcounter\@captype
     \expandafter\@firstofone
   \fi
   {\@dblarg{\@caption\@captype}}%
}
```

`\@caption`

```
\long\def\@caption#1[#2]#3{%
  \par
  \addcontentsline{\csname ext@#1\endcsname}{#1}%
    {\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
  \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 LaTeX constructs test for this flag and do not add vertical space when it is true. The intention is that this emulates TeX'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.

```
    \@parboxrestore
    \if@minipage
      \@setminipage
    \fi
```

```
    \normalsize
    \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
  \endgroup}
```

`\@float`

`\@dblflset`

```
\def\@float#1{%
  \@ifnextchar[%
    {\@xfloat{#1}}%
    {\edef\reserved@a{\noexpand\@xfloat{#1}[\csname fps@#1\endcsname]}%
     \reserved@a}}
```

`\@dblfloat`

```
\def\@dblfloat{%
  \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
  \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.

```
\def\@xfloat #1[#2]{%
  \@nodocument
  \def \@captype {#1}%
   \def \@fps {#2}%
   \@onelevel@sanitize \@fps
   \def \reserved@b {!}%
   \ifx \reserved@b \@fps
     \@fpsadddefault
   \else
     \ifx \@fps \@empty
       \@fpsadddefault
     \fi
   \fi
   \ifhmode
     \@bsphack
     \@floatpenalty -\@Mii
   \else
     \@floatpenalty-\@Miii
   \fi
  \ifinner
```

```
     \@parmoderr\@floatpenalty\z@
  \else
    \@next\@currbox\@freelist
      {%
       \@tempcnta \sixt@@n
       \expandafter \@tfor \expandafter \reserved@a
         \expandafter :\expandafter =\@fps
         \do
          {%
           \if \reserved@a h%
             \ifodd \@tempcnta
             \else
               \advance \@tempcnta \@ne
             \fi
           \fi
           \if \reserved@a t%
             \@setfpsbit \tw@
           \fi
           \if \reserved@a b%
             \@setfpsbit 4%
           \fi
           \if \reserved@a p%
             \@setfpsbit 8%
           \fi
           \if \reserved@a !%
             \ifnum \@tempcnta>15
               \advance\@tempcnta -\sixt@@n\relax
             \fi
           \fi
           }%
       \@tempcntb \csname ftype@\@captype \endcsname
       \multiply \@tempcntb \@xxxii
       \advance \@tempcnta \@tempcntb
       \global \count\@currbox \@tempcnta
       }%
    \@fltovf
  \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.

```
  \global \setbox\@currbox
    \color@vbox
      \normalcolor
      \vbox \bgroup
        \hsize\columnwidth
        \@parboxrestore
        \@floatboxreset
}
```

`\@floatboxreset` The rational 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.

```
\def \@floatboxreset {%
        \reset@font
        \normalsize
        \@setminipage
}
```

`\@setnobreak`

```
\def \@setnobreak{%
  \if@nobreak
    \let\outer@nobreak\@nobreaktrue
    \@nobreakfalse
  \fi
}
```

`\@setminipage`

```
\def \@setminipage{%
  \@minipagetrue
  \everypar{\@minipagefalse\everypar{}}%
}
```

`\end@float`

```
\def\end@float{%
  \@endfloatbox
  \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed `\textheight`, otherwise float will never get typeset (91/03/15 FMi).

```
    \@largefloatcheck
    \@cons\@currlist\@currbox
    \ifnum\@floatpenalty <-\@Mii
      \penalty -\@Miv
```

Saving and restoring `\prevdepth` added 26 May 87 to prevent extra vertical space when used in vertical mode.

```
      \@tempdima\prevdepth
      \vbox{}%
      \prevdepth\@tempdima
      \penalty\@floatpenalty

    \else
      \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
    \fi
  \fi
}
```

`\end@dblfloat`

```
\def\end@dblfloat{%
\if@twocolumn
  \@endfloatbox
  \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed `\textheight`, otherwise float will never get typeset (91/03/15 FMi).

```
    \@largefloatcheck
    \@cons\@dbldeferlist\@currbox
  \fi
```

RmS 92/03/18 changed `\@esphack` to `\@Esphack`.

```
    \ifnum \@floatpenalty =-\@Mii \@Esphack\fi
\else
   \end@float
\fi
}
```

`\@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.

```
\def \@endfloatbox{%
      \par\vskip\z@skip      %% \par\vskip\z@ added 15 Dec 87

      \@minipagefalse
      \outer@nobreak
    \egroup                  %% end of vbox
  \color@endbox
}
%
% \begin{macro}{\outer@nobreak}
% \changes{v1.0h}{1994/05/20}{Macro added: default is to do nothing.}
%    \begin{macrocode}
\let\outer@nobreak\@empty
```

`\@largefloatcheck` This calculates by how much a float is oversize for the page and prints this in a warning message.

```
\def \@largefloatcheck{%
  \ifdim \ht\@currbox>\textheight
    \@tempdima -\textheight
    \advance \@tempdima \ht\@currbox

    \@latex@warning {Float too large for page by \the\@tempdima}%
    \ht\@currbox \textheight
  \fi
}
```

`\@dbflt`

`\@xdblfloat`

```
\def\@dbflt#1{\@ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
\def\@xdblfloat#1[#2]{%
  \@xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}
```

Moved to ltoutput 93/12/16

```
%\newcount\c@topnumber
%\newcount\c@dbltopnumber
%\newcount\c@bottomnumber
%\newcount\c@totalnumber
```

An analysis of `\@floatplacement`:
This should be called whenever `\@colht` has been set.

```
\def\@floatplacement{\global\@topnum\c@topnumber
    % Textpage bit, global:
   \global\@toproom \topfraction\@colht
   \global\@botnum  \c@bottomnumber
   \global\@botroom \bottomfraction\@colht
   \global\@colnum  \c@totalnumber
    % Floatpage bit, local:
   \@fpmin   \floatpagefraction\@colht}
```

`\@dblfloatplacement` This should be called only within a group. Now changed to provide extra checks in `\@addtodblcol`, needed when processing a BANG float.

```
\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
  \global \@dbltopnum \c@dbltopnumber
  \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses `\@textmin` to locally store the amount of extra room in the column.

```
  \@textmin \@colht
  \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
  \@fpmin \dblfloatpagefraction\textheight
  \@fptop \@dblfptop
  \@fpsep \@dblfpsep
  \@fpbot \@dblfpbot
}
```

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., `\@leftmarginskip` == `\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`

```
\def\marginpar{%
  \ifhmode
    \@bsphack
    \@floatpenalty -\@Mii
  \else
    \@floatpenalty-\@Miii
  \fi
  \ifinner
    \@parmoderr
    \@floatpenalty\z@
  \else
    \@next\@currbox\@freelist{}{}%
    \@next\@marbox\@freelist{\global\count\@marbox\m@ne}%
       {\@floatpenalty\z@
        \@fltovf\def\@currbox{\@tempboxa}\def\@marbox{\@tempboxa}}%
  \fi
  \@ifnextchar [\@xmpar\@ympar}
```

`\@xmpar`

```
\long\def\@xmpar[#1]#2{%
  \@savemarbox\@marbox{#1}%
  \@savemarbox\@currbox{#2}%
  \@xympar}
```

`\@ympar`

```
\long\def\@ympar#1{%
  \@savemarbox\@marbox{#1}%
  \global\setbox\@currbox\copy\@marbox
  \@xympar}
```

`\@savemarbox`

```
\long\def \@savemarbox #1#2{%
  \global\setbox #1%
    \color@vbox
      \vtop{%
        \hsize\marginparwidth
        \@parboxrestore
        \@marginparreset
        #2%
        \@minipagefalse
        \outer@nobreak
        }%
    \color@endbox
}
```

`\@marginparreset` The rational 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.

```
\def \@marginparreset {%
        \reset@font
        \normalsize
%        \let\if@nobreak\iffalse
%        \let\if@noskipsec\iffalse
%        \@setnobreak
        \@setminipage
}
```

`\@xympar`

Setting the box here is done only because the code uses `\end@float`; it will be empty and gets discarded.

```
\def \@xympar{%
  \ifnum\@floatpenalty <\z@\@cons\@currlist\@marbox\fi
  \setbox\@tempboxa
    \color@vbox
      \vbox \bgroup
  \end@float
  \@ignorefalse
  \@esphack
}
```

`\reversemarginpar`
`\normalmarginpar`

```
\def\reversemarginpar{\global\@mparbottom\z@ \@reversemargintrue}
\def\normalmarginpar{\global\@mparbottom\z@ \@reversemarginfalse}
```

```
\message{footnotes,}
```

62.2 Footnotes

`\footnote`{NOTE} : User command to insert a footnote.

`\footnote`[NUM]{NOTE}: User command to insert a footnote numbered NUM, where NUM is a number – 1, 2, etc. For example, if footnotes are numbered *, **, etc. within pages, then `\footnote`[2]{...} produces footnote '**'. This command does not step the footnote counter.

`\footnotemark`[NUM] : Command to produce just the footnote mark in the text, but no footnote. With no argument, it steps the footnote counter before generating the mark.

`\footnotetext`[NUM]{TEXT} : Command to produce the footnote but no mark. `\footnote` is equivalent to `\footnotemark \footnotetext` .

As in PLAIN, footnotes use `\insert\footins`, and the following parameters:

`\footnotesize` : Size-changing command for footnotes.

`\footnotesep` : The height of a strut placed at the beginning of every footnote.

`\skip\footins` : Space between main text and footnotes. The rule
separating footnotes from text occurs in this
space. This space lies above the strut of height
`\footnotesep` which is at the beginning of the
first footnote.

`\footnoterule` : Macro to draw the rule separating footnotes from
text. It is executed right after a `\vspace` of
`\skip\footins.` It should take zero vertical
space–i.e., it should to a negative skip to
compensate for any positive space it occupies.
(See PLAIN.TEX.)

`\interfootnotelinepenalty` : Interline penalty for footnotes.

`\thefootnote` : In usual LaTeX style, produces the footnote number.
If footnotes are to be numbered within pages, then
the document style file must include an `\@addtoreset`
command to cause the footnote counter to be reset
when the page counter is stepped. This is not a good
idea, though, because the counter will not always be
reset in time to ensure that the first footnote on a
page is footnote number one.

`\@thefnmark` : Holds the current footnote's mark–e.g., `\dag` or '1'
or 'a'.

`\@mpfnnumber` : A macro that generates the numbers for `\footnote`
and `\footnotemark` commands. It == `\thefootnote`
outside a minipage environment, but can be
changed inside to generate numbers for
`\footnote`'s.

`\@makefnmark` : A macro to generate the footnote marker from
`\@thefnmark` The default definition was
`\hbox{$^\@thefnmark$}`.

This is now replaced by
`\textsuperscript{\@thefnmark}`

`\@makefntext{`NOTE`}` :
Must produce the actual footnote, using `\@thefnmark` as the mark
of the footnote and NOTE as the text. It is called when
effectively inside a `\parbox,` with `\hsize` = `\columnwidth.`
For example, it might be as simple as
`$^{\@thefnmark}$` NOTE

In a minipage environment, `\footnote` and `\footnotetext` are redefined
so that
(a) they use the counter mpfootnote
(b) the footnotes they produce go at the bottom of the minipage.
The switch is accomplished by letting `\@mpfn` == footnote or mpfootnote
and `\thempfn` == `\thefootnote` or `\thempfootnote,` and by redefining
`\@footnotetext` to be `\@mpfootnotetext` in the minipage.

`\footnote{`NOTE`}` ==
BEGIN

```
    \stepcounter{\@mpfn}
    begingroup
       \protect == \noexpand
       \@thefnmark :=G eval (\thempfn)
    endgroup
    \@footnotemark
    \@footnotetext{NOTE}
  END

\footnote[NUM]{NOTE} ==
  BEGIN
    begingroup
       \protect == \noexpand
       counter \@mpfn :=L NUM
       \@thefnmark :=G eval (\thempfn)
    endgroup
    \@footnotemark
    \@footnotetext{NOTE}
  END

\footnotemark       ==
  BEGIN \stepcounter{footnote}
        begingroup
           \protect == \noexpand
           \@thefnmark :=G eval(\thefootnote)
        endgroup
        \@footnotemark
  END

\footnotemark[NUM] ==
   BEGIN
       begingroup
         footnote counter :=L NUM
         \protect == \noexpand
        \@thefnmark :=G eval(\thefootnote)
       endgroup
       \@footnotemark
   END

\@footnotemark ==
   BEGIN
    \leavevmode
    IF hmode THEN \@x@sf := \the\spacefactor FI
    \@makefnmark           % put number in main text
    IF hmode THEN \spacefactor := \@x@sf FI
   END

\footnotetext       ==
    BEGIN begingroup \protect == \noexpand
                     \@thefnmark :=G eval (\thempfn)
          endgroup
          \@footnotetext
    END

\footnotetext[NUM] ==
    BEGIN begingroup  counter \@mpfn :=L NUM
                      \protect == \noexpand
```

```
                    \@thefnmark :=G eval (\thempfn)
            endgroup
            \@footnotetext
        END
```

`\footins` LaTeX does use the same insert for footnotes as PLAIN.

```
\newinsert\footins
```

LaTeX leaves these initializations for the `\footins` insert.

```
\skip\footins=\bigskipamount % space added when footnote is present
\count\footins=1000 % footnote magnification factor (1 to 1)
\dimen\footins=8in % maximum footnotes per page
```

`\footnoterule` LaTeX keeps PLAIN TeX's `\footnoterule` as the default.

```
\def\footnoterule{\kern-3\p@
  \hrule \@width 2in \kern 2.6\p@} % the \hrule is .4pt high
```

`\thefootnote`

```
\@definecounter{footnote}
\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.

```
\@definecounter{mpfootnote}
\def\thempfootnote{{\itshape\@alph\c@mpfootnote}}
```

`\@makefnmark` Default definition.

```
%\def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
```

`\textsuperscript` This command provides superscript characters in the current text font. It's implementation might change!!!

```
\DeclareRobustCommand*\textsuperscript[1]{%
  \@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.

```
\def\@textsuperscript#1{%
  {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}}
```

`\footnotesep`

```
\newdimen\footnotesep
```

`\footnote`

```
\def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark\@footnotetext}}
```

`\@xfootnote`

```
\def\@xfootnote[#1]{%
  \begingroup
    \csname c@\@mpfn\endcsname #1\relax
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotemark\@footnotetext}
```

\@footnotetext

```
\long\def\@footnotetext#1{\insert\footins{%
    \reset@font\footnotesize
    \interlinepenalty\interfootnotelinepenalty
    \splittopskip\footnotesep
    \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
    \hsize\columnwidth \@parboxrestore
    \protected@edef\@currentlabel{%
       \csname p@footnote\endcsname\@thefnmark
    }%
    \color@begingroup
      \@makefntext{%
        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
    \color@endgroup}}%
```

\footnotemark

```
\def\footnotemark{%
   \@ifnextchar[\@xfootnotemark
     {\stepcounter{footnote}%
      \protected@xdef\@thefnmark{\thefootnote}%
      \@footnotemark}}
```

\@xfootnotemark

```
\def\@xfootnotemark[#1]{%
   \begingroup
      \c@footnote #1\relax
      \unrestored@protected@xdef\@thefnmark{\thefootnote}%
   \endgroup
   \@footnotemark}
```

\@footnotemark

```
\def\@footnotemark{%
  \leavevmode
  \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
  \@makefnmark
  \ifhmode\spacefactor\@x@sf\fi
  \relax}
```

\footnotetext

```
\def\footnotetext{%
     \@ifnextchar [\@xfootnotenext
       {\protected@xdef\@thefnmark{\thempfn}%
    \@footnotetext}}
```

\@xfootnotenext

```
\def\@xfootnotenext[#1]{%
  \begingroup
     \csname c@\@mpfn\endcsname #1\relax
     \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotetext}
```

\thempfn

\@mpfn

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
⟨/2ekernel⟩
```

File H

ltidxglo.dtx

63 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
                             \beginggroup
                                \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.

```
⟨*2ekernel⟩
\message{index,}
```

`\makeindex`

```
\def\makeindex{%
  \newwrite\@indexfile
  \immediate\openout\@indexfile=\jobname.idx
  \def\index{\@bsphack\begingroup
             \@sanitize
             \@wrindex}\typeout
    {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.

```
  \let\makeindex\@empty
}
\@onlypreamble\makeindex
```

`\@wrindex`

```
\def\@wrindex#1{%
    \protected@write\@indexfile{}%
      {\string\indexentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

`\index`

```
\def\index{\@bsphack\begingroup \@sanitize\@index}
```

`\@index`

```
\def\@index#1{\endgroup\@esphack}
```

`\makeglossary`

```
\def\makeglossary{%
  \newwrite\@glossaryfile
  \immediate\openout\@glossaryfile=\jobname.glo
  \def\glossary{\@bsphack\begingroup
                \@sanitize
                \@wrglossary}\typeout
    {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.

```
  \let\makeglossary\@empty
}
\@onlypreamble\makeglossary
```

`\@wrglossary`

```
\def\@wrglossary#1{%
   \protected@write\@glossaryfile{}%
      {\string\glossaryentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

`\glossary`

```
\def\glossary{\@bsphack\begingroup\@sanitize\@index}
```

```
⟨/2ekernel⟩
```

File I

ltbibl.dtx

64 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 bibiography.

```
⟨*2ekernel⟩
\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`

```
\def\bibitem{\@ifnextchar[\@lbibitem\@bibitem}
```

`\@lbibitem`

```
\def\@lbibitem[#1]#2{\item[\@biblabel{#1}\hfill]\if@filesw
      {\let\protect\noexpand
       \immediate
       \write\@auxout{\string\bibcite{#2}{#1}}}\fi\ignorespaces}
```

`\@bibitem`

```
\def\@bibitem#1{\item\if@filesw \immediate\write\@auxout
       {\string\bibcite{#1}{\the\value{\@listctr}}}\fi\ignorespaces}
```

`\bibcite`

```
\def\bibcite{\@newl@bel b}
```

`\citation`

```
\let\citation\@gobble
```

`\cite`

```
\DeclareRobustCommand\cite{%
  \@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)

```
\def\@citex[#1]#2{\leavevmode
  \let\@citea\@empty
  \@cite{\@for\@citeb:=#2\do
    {\@citea\def\@citea{,\penalty\@m\ }%
     \edef\@citeb{\expandafter\@firstofone\@citeb\@empty}%
     \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.

```
     \@ifundefined{b@\@citeb}{\hbox{\reset@font\bfseries ?}%
       \G@refundefinedtrue
       \@latex@warning
         {Citation `\@citeb' on page \thepage \space undefined}}%
       {\@cite@ofmt{\csname b@\@citeb\endcsname}}}}{#1}}
```

`\bibdata`
`\bibstyle`

```
\let\bibdata=\@gobble
\let\bibstyle=\@gobble
```

\bibliography

```
\def\bibliography#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibdata{#1}}%
  \fi
  \@input@{\jobname.bbl}}
```

\bibliographystyle

```
\def\bibliographystyle#1{%
  \ifx\@begindocumenthook\@undefined\else
    \expandafter\AtBeginDocument
  \fi
    {\if@filesw
       \immediate\write\@auxout{\string\bibstyle{#1}}%
     \fi}}
```

\nocite (Added 14 Jun 85)

This puts information on the `.aux` file that causes BibTeX to include the citation list in the bibliography, but puts nothing in the text.

RmS 93/08/06: Made loop for `\nocite` like that for `\@citex`, to get rid of leading spaces.

```
\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 2_ε successor.

```
  \ifx\@onlypreamble\document
```

Since we are after `\begin{document}` we can do the citations:

```
    \@for\@citeb:=#1\do{%
      \edef\@citeb{\expandafter\@firstofone\@citeb}%
      \if@filesw\immediate\write\@auxout{\string\citation{\@citeb}}\fi
      \@ifundefined{b@\@citeb}{\G@refundefinedtrue
          \@latex@warning{Citation `\@citeb' undefined}}{}}%
  \else
```

But before `\begin{document}` we raise an error message:

```
    \@latex@error{Cannot be used in preamble}\@eha
```

Without the compatibility problems we could fix the problem as follows:

```
    % \AtBeginDocument{\nocite{#1}}
  \fi
  \@esphack}
```

Since `\nocite{*}` should not produce a warning about undefined citation keys (seee 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).

```
\expandafter\let\csname b@*\endcsname\@empty
```

64.1 Default definitions

This hook determines the 'relative formatting' of the two logical parts of a citation with comment.

\@cite

```
\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.

```
\let\@cite@ofmt\hbox
```

`\@biblabel`

```
\def\@biblabel#1{[#1]}
⟨/2ekernel⟩
```

File J

ltpage.dtx

65 Page styles and related commands

65.1 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.

65.2 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.)

65.3 marking conventions

LaTeX extends TeX'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 TeX's `\botmark`.
`\rightmark` : Used in the output routine, gets the current 'right' mark. Works like TeX'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.

```
⟨*2ekernel⟩
```

`\pagestyle` User command to set the page style for this and following pages.

```
\def\pagestyle#1{%
  \@ifundefined{ps@#1}%
    \undefinedpagestyle
    {\@nameuse{ps@#1}}}
```

`\thispagestyle` User command to set the page style for this page only.

```
\def\thispagestyle#1{%
  \@ifundefined{ps@#1}%
    \undefinedpagestyle
    {\global\@specialpagetrue\gdef\@specialstyle{#1}}}
```

`\ps@empty` The empty page style: No head or foot line.

```
\def\ps@empty{%
  \let\@mkboth\@gobbletwo\let\@oddhead\@empty\let\@oddfoot\@empty
  \let\@evenhead\@empty\let\@evenfoot\@empty}
```

`\ps@plain` The plain page style: No head, centred page number in foot.

```
\def\ps@plain{\let\@mkboth\@gobbletwo
      \let\@oddhead\@empty\def\@oddfoot{\reset@font\hfil\thepage
      \hfil}\let\@evenhead\@empty\let\@evenfoot\@oddfoot}
```

`\@leftmark` `\@rightmark` 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.

```
\let\@leftmark\@firstoftwo
\let\@rightmark\@secondoftwo
```

`\markboth` `\markright` User commands for setting LaTeX marks.

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`

```
\def\markboth#1#2{%
  \begingroup
    \let\label\relax \let\index\relax \let\glossary\relax
    \unrestored@protected@xdef\@themark {{#1}{#2}}%
    \@temptokena \expandafter{\@themark}%
    \mark{\the\@temptokena}%
  \endgroup
  \if@nobreak\ifvmode\nobreak\fi\fi}
\def\markright#1{%
  \begingroup
    \let\label\relax \let\index\relax \let\glossary\relax
```

Protection is handled inside `\@markright`.

```
    \expandafter\@markright\@themark {#1}%
    \@temptokena \expandafter{\@themark}%
    \mark{\the\@temptokena}%
  \endgroup
  \if@nobreak\ifvmode\nobreak\fi\fi}
```

`\@markright` `\leftmark` `\rightmark`

```
\def\@markright#1#2#3{\@temptokena {#1}%
  \unrestored@protected@xdef\@themark{{\the\@temptokena}{#3}}}
\def\leftmark{\expandafter\@leftmark\botmark\@empty\@empty}
\def\rightmark{\expandafter\@rightmark\firstmark\@empty\@empty}
```

`\@themark` Initialise LaTeX's marks without setting a TeX mark ⟨*whatsit*⟩.

```
\def\@themark{{}{}}
```

`\mark` Test versions of LaTeX 2_ε initialised TeX's `\mark` system at this point, but this was removed before the first release.

```
\AtBeginDocument{\mark{{}{}}}
```

`\raggedbottom` `\raggedbottom` typesets pages with no vertical stretch, so they have their natural height instead of all being exactly the same height. (Uses a space of .0001fil to avoid interfering with the 1fil space of `\newpage`.)

```
\def\raggedbottom{%
  \def\@textbottom{\vskip \z@ \@plus.0001fil}\let\@texttop\relax}
```

`\flushbottom` `\flushbottom`: Inverse of `\raggedbottom` — makes all pages the same height.

```
\def\flushbottom{%
  \let\@textbottom\relax \let\@texttop\relax}
```

`\sloppy` `\sloppy` will never (well, hardly ever) produce overfull boxes, but may produce underfull ones. (14 June 85)

```
\def\sloppy{%
  \tolerance 9999%
  \emergencystretch 3em%
  \hfuzz .5\p@
  \vfuzz\hfuzz}
```

`sloppypar` A sloppypar environment is equivalent to `{\par \sloppy ... \par}`.

```
\def\sloppypar{\par\sloppy}
\def\endsloppypar{\par}
```

`\fussy` Resets TeX's parameters to their normal finicky values.

```
\def\fussy{%
  \emergencystretch\z@
  \tolerance 200%
  \hfuzz .1\p@
  \vfuzz\hfuzz}
```

`\overfullrule` LaTeX default is no overfull box rule. Changed by document class option.

```
\overfullrule 0pt
```

```
⟨/2ekernel⟩
```

File K

ltoutput.dtx

66 Output Routine

66.1 Floats

The '2ekernel' code ensures that a `\usepackage{autoout1}` is essentially ignored if a 'full' format is being used that has the autoload file mode already in the format.

```
⟨defx⟩\begingroup
⟨defx⟩\makeatletter
⟨defx⟩\nfss@catcodes
⟨2ekernel⟩\expandafter\let\csname ver@autoout1.sty\endcsname\fmtversion
```

```
⟨*2ekernel | autoload⟩
\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
`\@dblfptop, \@dblfpsep, \@dblfpbot`
: Analogous for double-column float page in two-column format.

FOOTNOTES: As in PLAIN, footnotes use `\insert\footins.`

PAGE LAYOUT SWITCHES AND MACROS

@twocolumn : Boolean. T if two columns per page globally.

PAGE STYLE MACROS AND SWITCHES

`\@oddhead` : IF @twoside = T
THEN macro to generate head of odd-numbered pages.
ELSE macro to generate head of all pages.
`\@evenhead` : IF @twoside = T
THEN macro to generate head of even-numbered pages.
`\@oddfoot` : IF @twoside = T
THEN macro to generate foot of odd-numbered

```
                          pages.
                     ELSE macro to generate foot of all pages.
  \@evenfoot        : IF @twoside = T
                     THEN macro to generate foot of
even-numbered
                          pages.
  @specialpage     : boolean.  T if current page is to have a special
                         format.
 \@specialstyle  : If its value is  foo then
                   IF @specialpage = T
                     THEN the command \ps@foo is executed to
                          temporarily reset the page style parameters
                          before composing the current page.
                          This command should execute only \def's
and
                          \edef's, making only local definitions.

 FLOAT PLACEMENT PARAMETERS

The following parameters are set by the macro \@floatplacement.
When \@floatplacement is called,
\@colht is the height of the page or column being built.  I.e.:
        * For single-column page it equals \textheight.
        * For double-column page it equals \textheight - height
          of double-column floats on page.
Note that some are set globally and some locally:
   \@topnum  :=G Maximum number of floats allowed on the top of a
                 column.
   \@toproom :=G Maximum amount of top of column devoted to floats–
                 excluding \textfloatsep separation below the floats
                 and \floatsep separation between them.  For
                 two-column output, should be computed as a function
                 of \@colht.
   \@botnum, \@botroom
               : Analogous to above.
   \@colnum  :=G Maximum number of floats allowed in a column,
                 including in-text floats.
   \@textmin :=L Minimum amount of text (excluding footnotes) that
                 must appear on a text page.
                    %% 27 Sep 85 : made local to
                    %% \@addtocurcol and \@addtonextcol
                 It is now also used locally in processing double
                 floats.
   \@fpmin   :=L Minimum height of floats in a float column.

The macro \@dblfloatplacement sets the following parameters.
   \@dbltopnum  :=G Maximum number of double-column floats allowed
at
                    the top of a two-column page.
   \@dbltoproom :=G Maximum height of double-column floats allowed at
                    top of two-column page.
   \@fpmin      :=L Minimum height of floats in a float column.
It should also perform the following local assignments where necessary
– i.e., where the new value differs from the old one:
      \@fptop      :=L \@dblfptop
      \@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	`\pagebreak` `\newpage`
-10001	`\clearpage` (`\penalty` -10000 `\vbox{}` `\penalty` -10001)
-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 incluses 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: log2 [(\count I)/32] is the bit number corresponding to the type of float I. To see if there is any float in \LIST having the same type as float I, you run \@bitor with
\NUM = [(\count I)/32] * 32.

```
\@bitor\NUM\LIST ==
  BEGIN
     @test :=G false
     { \@elt \CTR ==  if \NUM <> 0 then
                        if \count\CTR / \NUM is odd
                           then   @test := true         fi fi
       \LIST
     }
  END
```

```
\@cons\LIST\NUM : Globally sets \LIST := \LIST * \@elt \NUM

\@cons\LIST\NUM ==
  BEGIN {  \@elt == \relax
           \LIST :=G \LIST \@elt \NUM
        }
```

BOX LISTS FOR FLOAT-PLACEMENT ALGORITHMS

```
  \@freelist     : List of empty boxes for placing new floats.
  \@toplist      : List of floats to go at top of current column.
  \@midlist      : List of floats in middle of current column.
  \@botlist      : List of floats to go at bottom of current column.
  \@deferlist    : List of floats to go after current column.
  \@dbltoplist   : List of double-col. floats to go at top of current
                    page.
  \@dbldeferlist : List of double-column floats to go on subsequent
                    pages.
```

FLOAT-PLACEMENT ALGORITHMS

```
 \@addtobot : Tries to put insert \@currbox on \@botlist.
              Called only when:
                 * \ht BOX < \@colroom
                 * type of \@currbox not on \@deferlist
                 * \@colnum > 0
                 * @insert = false
              If it succeeds, then:
                 * sets @insert true
                 * decrements \@botroom by \ht BOX
                 * decrements \@botnum and \@colnum by 1
                 * decrements \@colroom by \ht BOX + either
\floatsep
                   or \textfloatsep, as appropriate.
                 * sets \maxdepth to 0pt

 \@addtotoporbot : Tries to put insert \@currbox on \@toplist or
                   \@botlist.
                   Called only under same conditions as \@addtobot.
```

If it succeeds, then:

- sets @insert true
- decrements `\@toproom` or `\@botroom` by `\ht` BOX
- decrements `\@colnum` and either `\@topnum` or `\@botnum` by 1
- decrements `\@colroom` by `\ht` BOX + `\floatsep` or `\textfloatsep`, as appropriate.

`\@addtocurcol` : Tries to add `\@currbox` to current column, setting @insert true if it succeeds, false otherwise. It will add `\@currbox` to top only if bit 0 of `\count \@currbox` is 0, and to the bottom only if bit 0 = 0 or an earlier float of the same type is put on the bottom. If the float is put in the text, then `\penalty\interlinepenalty` is put right after the float, before the following `\vskip`, and `\outputpenalty` :=L 0.

`\@addtonextcol` : Tries to add `\@currbox` to the next column, setting @insert true if it succeeds, false otherwise.

`\@addtodblcol` : Tries to add `\@currbox` to the next double-column page, adding it to `\@dbltoplist` if it succeeds and `\@dbldeferlist` if it fails.

```
\@addmarginpar ==
 BEGIN
   if \@currlist nonempty
     then remove \@marbox from \@currlist
          add \@marbox and \@currbox to \@freelist
               %% NOTE: \@currbox = left box
     else LaTeX error: ?  %% shouldn't happen
   fi
   \@tempcnta := 1      %% 1 = right, -1 = left
   if @twocolumn = true
     then if @firstcolumn = true
             then \@tempcnta := -1
          fi
     else if @mparswitch = true
             then if count0 odd
                     else \@tempcnta := -1
                  fi
          fi
          if @reversemargin = true
              then \@tempcnta := -\@tempcnta
          fi
   fi
   if \@tempcnta < 0 then \box\@marbox :=G \box\@currbox
   fi
   \@tempdima   :=L maximum(\@mparbottom - \@pageht
                                          + ht of \@marbox, 0)
   if \@tempdima > 0 then LaTeX warning: 'marginpar moved' fi
   \@mparbottom :=G \@pageht + \@tempdima + depth of \@marbox
```

```
                         + \marginparpush
    \@tempdima   :=L \@tempdima - ht of \@marbox
    \box\@marbox :=G \box\@currbox
                               \vbox { \vskip \@tempdima
                                       \box\@marbox
                                      }
    height of \@marbox :=G depth of \@marbox :=G 0
    \kern -\@pagedp
    \nointerlineskip
    \hbox{ if @tempcnta > 0 then \hskip \columnwidth
                                 \hskip \marginparsep
                            else \hskip -\marginparsep
                                 \hskip -\marginparwidth
           fi
           \box\@marbox \hss
         }
    \nobreak
    \nointerlineskip
    \hbox{\vrule height 0 width 0 depth \@pagedp}
  END
```

Floats and marginpars add a lot of dead cycles.

```
\maxdeadcycles = 100
```

```
\let\@elt\relax
```

```
\def\@next#1#2#3#4{\ifx#2\@empty #4\else
   \expandafter\@xnext #2\@@#1#2#3\fi}
```

```
\def\@xnext \@elt #1#2\@@#3#4{\def#3{#1}\gdef#4{#2}}
```

\changes{v1.1v}{1996/07/26}{put \cs{global} into definition}

```
\def\@testfalse{\global\let\if@test\iffalse}
\def\@testtrue {\global\let\if@test\iftrue}
\@testfalse
```

\changes{v1.1v}{1996/07/26}{remove \cs{global} before \cs{@test...}}

```
\def\@bitor#1#2{\@testfalse {\let\@elt\@xbitor
   \@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...}}

```
\def\@xbitor #1{\@tempcntb \count#1
   \ifnum \@tempcnta =\z@
   \else
     \divide\@tempcntb\@tempcnta
     \ifodd\@tempcntb \@testtrue\fi
   \fi}
```

DEFINITION OF FLOAT BOXES:

```
\newinsert\bx@A
\newinsert\bx@B
\newinsert\bx@C
\newinsert\bx@D
\newinsert\bx@E
\newinsert\bx@F
\newinsert\bx@G
\newinsert\bx@H
```

```
\newinsert\bx@I
\newinsert\bx@J
\newinsert\bx@K
\newinsert\bx@L
\newinsert\bx@M
\newinsert\bx@N
\newinsert\bx@O
\newinsert\bx@P
\newinsert\bx@Q
\newinsert\bx@R
```

```
\gdef\@freelist{\@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
               \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
                \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
                \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
```

```
\gdef\@toplist{}
\gdef\@botlist{}
\gdef\@midlist{}
\gdef\@currlist{}
\gdef\@deferlist{}
\gdef\@dbltoplist{}
\gdef\@dbldeferlist{}
```

PAGE LAYOUT PARAMETERS

```
\newdimen\topmargin
\newdimen\oddsidemargin
\newdimen\evensidemargin
\let\@themargin=\oddsidemargin
\newdimen\headheight
\newdimen\headsep
\newdimen\footskip
\newdimen\textheight
\newdimen\textwidth
\newdimen\columnwidth
\newdimen\columnsep
\newdimen\columnseprule
\newdimen\marginparwidth
\newdimen\marginparsep
\newdimen\marginparpush
```

`\AtBeginDvi` `\@begindvibox`

We use a box register in which to put stuff that must appear before anything else in the `.dvi` file.

The stuff in the box should not add any typeset material to the page when it is unboxed.

```
\newbox\@begindvibox
\def \AtBeginDvi #1{%
  \global \setbox \@begindvibox
    \vbox{\unvbox \@begindvibox #1}%
}
```

`\@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?

```
\newdimen\@maxdepth
\@maxdepth = \maxdepth
```

`\paperheight` `\paperwidth`

New `\paper`... registers.

```
\newdimen\paperheight
\newdimen\paperwidth
```

`\if@insert` `\if@fcolmade` `\if@specialpage` `\if@firstcolumn` `\if@twocolumn` `\if@twoside` `\if@reversemarginpar` `\if@mparswitch` `\col@number`

Local switches first:

```
\newif \if@insert
```

These should definitely be global:

```
\newif \if@fcolmade
\newif \if@specialpage \@specialpagefalse
```

These should be global but are not always set globally in other files.

```
\newif \if@firstcolumn \@firstcolumntrue
\newif \if@twocolumn   \@twocolumnfalse
```

Not sure about these: two questions. Should things which must apply to a whole doument be local or global (they probably should be 'preamble only' commands)? Are these three such things?

```
\newif \if@twoside     \@twosidefalse
\newif \if@reversemargin \@reversemarginfalse
\newif \if@mparswitch  \@mparswitchfalse
```

This counter has been imported from 'multicol'.

```
\newcount \col@number
\col@number \@ne
```

INTERNAL REGISTERS

```
\newcount\@topnum
\newdimen\@toproom
\newcount\@dbltopnum
\newdimen\@dbltoproom
\newcount\@botnum
\newdimen\@botroom
\newcount\@colnum
\newdimen\@textmin
\newdimen\@fpmin
\newdimen\@colht
\newdimen\@colroom
\newdimen\@pageht
\newdimen\@pagedp
\newdimen\@mparbottom \@mparbottom\z@
\newcount\@currtype
\newbox\@outputbox
\newbox\@leftcolumn
\newbox\@holdpg
```

```
\def\@thehead{\@oddhead} % initialization
\def\@thefoot{\@oddfoot}
```

`\clearpage` The tests at the beginning are an experimental attempt to avoid a completely empty page after a `\twocolumn[...]`. This prevents the text from the argument vanishing into a float box, never to be seen again. We hope that it does not produce wrong formatting in other cases.

```
\def\clearpage{%
  \ifvmode
    \ifnum \@dbltopnum =\m@ne
      \ifdim \pagetotal <\topskip
        \hbox{}%
      \fi
    \fi
  \fi
  \newpage
  \write\m@ne{}%
  \vbox{}%
  \penalty -\@Mi
}
```

`\cleardoublepage`

```
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
    \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
⟨/2ekernel | autoload⟩
```

`\onecolumn`

```
⟨*2ekernel | autoload | fltrace⟩
\def\onecolumn{%
  \clearpage
  \global\columnwidth\textwidth
  \global\hsize\columnwidth
  \global\linewidth\columnwidth
  \global\@twocolumnfalse
  \col@number \@ne
  \@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`.

```
\def \newpage {%
  \if@noskipsec
    \ifx \@nodocument\relax
      \leavevmode
      \global \@noskipsecfalse
    \fi
  \fi
  \if@inlabel
    \leavevmode
    \global \@inlabelfalse
  \fi
  \if@nobreak \@nobreakfalse \everypar{}\fi
  \par
  \vfil
  \penalty -\@M}
```

`\@emptycol` It may be better to use an invisible rule rather than an empty box here.

```
\def \@emptycol {\vbox{}\penalty -\@M}
```

`\twocolumn` There are several bug fixes to the two-column stuff here.

`\@topnewpage`

```
\def \twocolumn {%
  \clearpage
  \global\columnwidth\textwidth
  \global\advance\columnwidth-\columnsep
  \global\divide\columnwidth\tw@
  \global\hsize\columnwidth
  \global\linewidth\columnwidth
  \global\@twocolumntrue
  \global\@firstcolumntrue
  \col@number \tw@
```

There is no reason to put a `\@dblfloatplacement` here since `\@topnewpage` ignores these settings. The `\@floatplacement` is needed in case this comes after some changes.

```
  \@ifnextchar [\@topnewpage\@floatplacement
}
```

Note that here, getting a box from the freelist can assume success since this comes just after a `\clearpage`.

```
\long\def \@topnewpage [#1]{%
```

```
\@nodocument
\@next\@currbox\@freelist{}{}%
\global \setbox\@currbox
  \color@vbox
    \normalcolor
    \vbox {%
      \hsize\textwidth
      \@parboxrestore
      \col@number \@ne
      #1%
      \vskip -\dbltextfloatsep
           }%
  \color@endbox
```

Added size test and warning message; perhaps we should use an error message.

```
\ifdim \ht\@currbox>\textheight
  \ht\@currbox \textheight
\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.

```
\global \count\@currbox \tw@
\@tempdima -\ht\@currbox
\advance \@tempdima -\dbltextfloatsep
\global \advance \@colht \@tempdima
\ifx \@dbltoplist \@empty
\else
  \@latexerr{Float(s) lost}\@ehb
  \let \@dbltoplist \@empty
\fi
\@cons \@dbltoplist \@currbox
```

This setting of `\@dbltopnum` is used only to change the typesetting in `\@combinedblfloats`.

```
\global \@dbltopnum \m@ne
⟨*trace⟩
  \tr@ce{dbltopnum set to -1 (= \the \@dbltopnum) (topnewpage)}%
⟨/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.

```
\ifdim \@colht<2.5\baselineskip
  \@latex@warning@no@line {Optional argument of \noexpand\twocolumn
              too tall on page \thepage}%
  \@emptycol
```

```
    \if@firstcolumn
    \else
      \@emptycol
    \fi
  \else
    \global \vsize \@colht
    \global \@colroom \@colht
    \@floatplacement
  \fi
}
```

`\output`
`\@specialoutput`

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.

```
\output {%
  \let \par \@@par
  \ifnum \outputpenalty<-\@M
    \@specialoutput
  \else
    \@makecol
    \@opcol
```

Moved to `\@opcol`: `\@floatplacement`.

```
    \@startcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startcolumn`.

```
    \@whilesw \if@fcolmade \fi
      {%
⟨*trace⟩
       \tr@ce{PAGE: float \if@twocolumn column \else page \fi
                    completed}%
⟨/trace⟩
       \@opcol\@startcolumn}%
  \fi
  \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`.

```
    \ifdim \@colroom<1.5\baselineskip
      \ifdim \@colroom<\textheight
        \@latex@warning@no@line {Text page \thepage\space
                               contains only floats}%
        \@emptycol
```

```
%         \if@twocolumn
%           \if@firstcolumn
%           \else
%             \@emptycol
%           \fi
%         \fi
      \else
        \global \vsize \@colroom
      \fi
    \else
      \global \vsize \@colroom
    \fi
  \else
    \global \vsize \maxdimen
  \fi
}
⟨/2ekernel | autoload | fltrace⟩
```

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.)

```
⟨*2ekernel | def1 | autoload | fltrace⟩
\gdef\@specialoutput{%
   \ifnum \outputpenalty>-\@Mii
     \@doclearpage
   \else
     \ifnum \outputpenalty<-\@Miii
       \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
       \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
     \else
```

Note that `\boxmaxdepth` should not be set here since we wish to record the natural depth of the holdpg box.

This is changed so as to not lose anything, such as writes and marks, which may get into box 255 and should be returned to the list. This should only happen when the first penalty in the mechanism is discarded and therefore `\@holdpg` should always be void in this case. This can happen because a penalty is discarded whenever there is no box on the list.

It was just: `\setbox\@tempboxa \box \@cclv`.

The last box which is removed is the box put there by the double-penalty mechanism. The `\unskip` then removes the `\topskip` which is put there since the box is the first on the page.

```
       \global \setbox\@holdpg \vbox{%
                      \unvbox\@holdpg
                      \unvbox\@cclv
```

We must now remove the box added by the float mechanism and the `\topskip` glue therefore added above it by TeX.

```
                      \setbox\@tempboxa \lastbox
                      \unskip
                                     }%
```

These two are needed as separate dimensions only by `\@addmarginpar`; for other purposes we put the whole size into `\@pageht` (see below).

```
       \@pagedp \dp\@holdpg
       \@pageht \ht\@holdpg
       \unvbox \@holdpg
```

```
      \@next\@currbox\@currlist{%
        \ifnum \count\@currbox>\z@
```

Putting the whole size into `\@pageht` (see above).

```
          \advance \@pageht \@pagedp
          \ifvoid\footins \else
            \advance \@pageht \ht\footins
            \advance \@pageht \skip\footins
            \advance \@pageht \dp\footins
          \fi
⟨*2ekernel | def1⟩
          \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?

```
            \ifdim \wd\@kludgeins=\z@
              \advance \@pageht \ht\@kludgeins
⟨*trace⟩
              \tr@ce {Extra size added: \the \ht\@kludgeins}%
⟨/trace⟩
            \fi
          \fi
⟨/2ekernel | def1⟩
```

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.

```
          \@reinserts
          \@addtocurcol
        \else
         \@reinserts
         \@addmarginpar
        \fi
        }\@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.

```
      \ifnum \outputpenalty<\z@
        \if@nobreak
         \nobreak
        \else
         \addpenalty \interlinepenalty
        \fi
      \fi
    \fi
  \fi
}
⟨/2ekernel | def1 | autoload | fltrace⟩
```

`\@doclearpage` This is a very much an emergency action, just dumping everything: footnotes first then floats. A more sophisticated version is needed; but even more urgent is a bug-free version (see, for example, pr/3528).

Also, it puts any left-over non-boxes (writes, specials, etc.) back after any float pages created: this is a very bad bug since, for example, a kludge insert will be in quite the wrong place and, worse, be irremovable and uncancelable.

```
⟨*2ekernel | autoload⟩
\def \@doclearpage {%
    \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.

```
       \ifvbox\@kludgeins
         {\setbox \@tempboxa \box \@kludgeins}%
⟨*trace⟩
         \tr@ce {kludgeins box made void}%
⟨/trace⟩
       \fi
       \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
       \setbox\@tempboxa\box\@cclv
       \xdef\@deferlist{\@toplist\@botlist\@deferlist}%

       \global \let \@toplist \@empty
       \global \let \@botlist \@empty
       \global \@colroom \@colht
       \ifx \@currlist\@empty
       \else
          \@latexerr{Float(s) lost}\@ehb

          \global \let \@currlist \@empty
       \fi
       \@makefcolumn\@deferlist
       \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
       \if@twocolumn
         \if@firstcolumn
           \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%

           \global \let \@dbltoplist \@empty
           \global \@colht \textheight
           \begingroup
              \@dblfloatplacement
              \@makefcolumn\@dbldeferlist
              \@whilesw\if@fcolmade \fi{\@outputpage
                                        \@makefcolumn\@dbldeferlist}%
           \endgroup
         \else
           \vbox{}\clearpage
         \fi
       \fi
     \else
       \setbox\@cclv\vbox{\box\@cclv\vfil}%
       \@makecol\@opcol
       \clearpage
     \fi
}
⟨/2ekernel | autoload⟩
```

`\@opcol` Several changes in detail here.

```
⟨*2ekernel | autoload | fltrace⟩
\def \@opcol {%
  \if@twocolumn
    \@outputdblcol
  \else
    \@outputpage
⟨*trace⟩
    \tr@ce{PAGE: one column (float? see above) page completed}%
⟨/trace⟩
```

Not needed since it comes after `\@outputpage`:

```
%   \global\@colht\textheight
  \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.

```
  \global \@mparbottom \z@ \global \@textfloatsheight \z@
  \@floatplacement
}
⟨/2ekernel | autoload | fltrace⟩
```

`\@makecol` We must rewrite this macro to alllow for variations in page-makeup required by changes in page-length.

This uses a different macro if a special-length column is being produced.

```
⟨*2ekernel | def1 | autoload⟩
\gdef \@makecol {%
   \ifvoid\footins
     \setbox\@outputbox \box\@cclv
   \else
     \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.

```
       \boxmaxdepth \@maxdepth

%       \@tempdima\dp\@cclv
       \unvbox \@cclv
%       \vskip-\@tempdima
       \vskip \skip\footins

       \color@begingroup
         \normalcolor
         \footnoterule
         \unvbox \footins
       \color@endgroup
       }%
   \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.

```
   \let\@elt\relax
   \xdef\@freelist{\@freelist\@midlist}%

   \global \let \@midlist \@empty
   \@combinefloats
```

The variations start here in case `\enlargethispage` has been used.

```
⟨*2ekernel | def1⟩
   \ifvbox\@kludgeins
     \@makespecialcolbox
   \else
⟨/2ekernel | def1⟩
```

This extra reboxing is only needed to add the `\@texttop` and `\@textbotttom` 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 adjustemnt 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 wthin `\@texttop` and `\@textbottom`, should it? Is it needed at all?

RmS 91/10/22: Replaced `\dimen128` by `\dimen@`.

```
   \setbox\@outputbox \vbox to\@colht {%
%       \boxmaxdepth \maxdepth                    %??
       \@texttop
       \dimen@ \dp\@outputbox
       \unvbox \@outputbox
       \vskip -\dimen@
       \@textbottom
       }%
⟨*2ekernel | def1⟩
   \fi
⟨/2ekernel | def1⟩
   \global \maxdepth \@maxdepth
}
⟨/2ekernel | def1 | autoload⟩
```

`\@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 expidite experimentation.

```
⟨*2ekernel | def1 | autoload⟩
\gdef \@reinserts{%
  \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
⟨+2ekernel | def1⟩  \ifvbox\@kludgeins\insert\@kludgeins
⟨+2ekernel | def1⟩                      {\unvbox\@kludgeins}\fi
}
⟨/2ekernel | def1 | autoload⟩
```

`\@makespecialcolbox` This implements certain variations in page-makeup.

```
⟨*2ekernel | def1 | fltrace⟩
\gdef \@makespecialcolbox {%
⟨*trace⟩
   \tr@ce{Kludgeins ht \the\ht\@kludgeins\space
                           dp \the\dp\@kludgeins\space
                           wd \the\wd\@kludgeins}%
⟨/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'.

```
   \setbox\@outputbox \vbox {%
     \@texttop
     \dimen@ \dp\@outputbox
     \unvbox\@outputbox
     \vskip-\dimen@
     }%
   \@tempdima \@colht
   \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 TeX3 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.

```
     \advance \@tempdima -\ht\@outputbox
     \advance \@tempdima \pageshrink
⟨*trace⟩
     \tr@ce {Natural ht of col: \the \ht\@outputbox}%
     \tr@ce {\string \@colht: \the \@colht}%
     \tr@ce {Pageshrink added: \the \pageshrink}%
     \tr@ce {Hence, space added: \the \@tempdima}%
⟨/trace⟩
     \setbox\@outputbox \vbox to \@colht {%
%      \boxmaxdepth \maxdepth
       \unvbox\@outputbox
       \vskip \@tempdima
       \@textbottom
       }%
```

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.

```
   \else
     \advance \@tempdima -\ht\@kludgeins
⟨*trace⟩
     \tr@ce {Natural ht of col: \the \ht\@outputbox}%
     \tr@ce {\string \@colht: \the \@colht}%
     \tr@ce {Extra size added: -\the \ht \@kludgeins}%
     \tr@ce {Hence, height of inner box: \the \@tempdima}%
     \tr@ce {Max? pageshrink available: \the \pageshrink}%
⟨/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.

```
     \setbox \@outputbox \vbox to \@colht {%
       \vbox to \@tempdima {%
         \unvbox\@outputbox
         \@textbottom}%
       \vss}%
   \fi
```

Finally we need to explicitly make the insert box void.

```
   {\setbox \@tempboxa \box \@kludgeins}%
⟨*trace⟩
     \tr@ce {kludgeins box made void}%
⟨/trace⟩
}
⟨/2ekernel | def1 | fltrace⟩
```

`\@texttop`
`\@textbottom`

These do nothing as a default.

```
⟨*2ekernel | autoload⟩
\let \@texttop \relax
\let \@textbottom \relax
```

`\@resetactivechars`
`\@activechar@info`

RmS 93/09/06: added hook to protect against certain active characters in the output routine. Default checks are for active space and end-of-line.

```
\def\@activechar@info #1{%
      \@latex@info@no@line {Active #1 character found while
                            output routine is active
                            \MessageBreak
                            This may be a bug in a package file
                            you are using}%
}
```

Do not put any spaces in this next bit!

```
\begingroup
\obeylines\obeyspaces%
\catcode`\'\active%
\gdef\@resetactivechars{%
\def^^M{\@activechar@info{EOL}\space}%
\def {\@activechar@info{space}\space}%
\let'\active@math@prime}%
\endgroup
```

`\@outputpage`
`\@shipoutsetup`
`\@writesetup`

The `\color@hbox` hooks here are used to avoid putting just a colour special into an otherwise empty box (in a header or footer). These boxes are often set to be completely empty and so adding a special produces a very underfull box message.

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 `\write`s 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 `\write`s because the aftergroup token list is only constructed when popping the save-stack, it is not expanded until after the shipout is completed.

Question: should things from an `\aftergroup` within the shipped out box be executed in the environment set up for the writes, or after it finishes?

A lot of this code has been in-lined tp prevent mis-use of internal commands as hooks.

```
\def\@outputpage{%
\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`.

```
  \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?

```
  \@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.

```
  \global\let\@@if@newlist\if@newlist
  \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
```

```
\@parboxrestore
```

. . . to here was in the command `\@writesetup`.

```
\shipout \vbox{%
  \set@typeset@protect
  \aftergroup \endgroup
  \aftergroup \set@typeset@protect
                              % correct? or just restore by ending
                              % the group?
```

This first bit has been moved inside the shipped out box.

Now the setup inside the shipped out box; this should conatin 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`.

```
\if@specialpage
  \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
\fi
\if@twoside
  \ifodd\count\z@ \let\@thehead\@oddhead \let\@thefoot\@oddfoot
       \let\@themargin\oddsidemargin
  \else \let\@thehead\@evenhead
     \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
  \fi
\fi
```

The rest was always inside the box.

RmS 91/08/15: aded this line:

```
\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.

```
\normalsize
```

Reset the space factors.

```
\normalsfcodes
```

Reset these here (previously reset separately for head and foot)

```
\let\label\@gobble
\let\index\@gobble
\let\glossary\@gobble
```

```
\baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
```

. . . to here was in the command `\@shipoutsetup`.

```
    \@begindvi
    \vskip \topmargin
    \moveright\@themargin \vbox {%
      \setbox\@tempboxa \vbox to\headheight{%
        \vfil
        \color@hbox
          \normalcolor
          \hb@xt@\textwidth{\@thehead}%
        \color@endbox
        }%                        %% 22 Feb 87
      \dp\@tempboxa \z@
      \box\@tempboxa
      \vskip \headsep
      \box\@outputbox
      \baselineskip \footskip
      \color@hbox
        \normalcolor
        \hb@xt@\textwidth{\@thefoot}%
      \color@endbox
      }%
    }%
```

`\endgroup` now inserted by `\aftergroup`

Restore `\if@newlist`

```
  \global\let\if@newlist\@@if@newlist
```

```
  \global \@colht \textheight
  \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.

```
  \let\firstmark\botmark
}
```

`\@begindvi` 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.

```
\def \@begindvi{%
  \unvbox \@begindvibox
  \global\let \@begindvi \@empty
}
```

`\@combinefloats` `\@cflt` `\@cflb` 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 it has been moved there; this also agrees with the original pseudcode.

```
\def \@combinefloats {%
%    \boxmaxdepth \maxdepth
    \ifx \@toplist\@empty \else \@cflt \fi
    \ifx \@botlist\@empty \else \@cflb \fi
}
```

```
\def \@cflt{%
    \let \@elt \@comflelt
    \setbox\@tempboxa \vbox{}%
    \@toplist
    \setbox\@outputbox \vbox{%
                             \boxmaxdepth \maxdepth
                             \unvbox\@tempboxa
                             \vskip -\floatsep
                             \topfigrule
                             \vskip \textfloatsep
```

```
                            \unvbox\@outputbox
                            }%
    \let\@elt\relax
    \xdef\@freelist{\@freelist\@toplist}%
    \global\let\@toplist\@empty
}
\def \@cflb {%
    \let\@elt\@comflelt
    \setbox\@tempboxa \vbox{}%
    \@botlist
    \setbox\@outputbox \vbox{%
                            \unvbox\@outputbox
                            \vskip \textfloatsep
                            \botfigrule
                            \unvbox\@tempboxa
                            \vskip -\floatsep
                            }%
    \let\@elt\relax
    \xdef\@freelist{\@freelist\@botlist}%
    \global \let \@botlist\@empty
}
```

`\@comflelt`
`\@comdblflelt`
`\@combinedblfloats`

```
\def\@comflelt#1{\setbox\@tempboxa
      \vbox{\unvbox\@tempboxa\box #1\vskip\floatsep}}
\def\@comdblflelt#1{\setbox\@tempboxa
      \vbox{\unvbox\@tempboxa\box #1\vskip\dblfloatsep}}
\def \@combinedblfloats{%
  \ifx \@dbltoplist \@empty
  \else
    \setbox\@tempboxa \vbox{}%
    \let \@elt \@comdblflelt
    \@dbltoplist
    \let \@elt \relax
    \xdef \@freelist {\@freelist\@dbltoplist}%
    \global\let \@dbltoplist \@empty
    \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.

```
      {%\boxmaxdepth\maxdepth   %% probably not needed, CAR
       \unvbox\@tempboxa\vskip-\dblfloatsep
```

Here we need different typesetting if the top float comes from `\@topnewpage`.

```
       \ifnum \@dbltopnum>\m@ne
         \dblfigrule
       \fi
       \vskip \dbltextfloatsep
       \box\@outputbox
       }%
  \fi
}
⟨/2ekernel | autoload⟩
```

`\@startcolumn`
`\@startdblcolumn`

We could combine (most of) these two into `\@startcol <list>`. Note that `\@xstartcol` was only used once (i.e. in `\@startcolumn`); it has therefore been removed. This is not quite as efficient but it now has the same structure as `\@startdblcolumn`.

The empty-list test has been moved to `\@tryfcolumn`.

```
⟨*2ekernel | autoload | fltrace⟩
```

```
\def \@startcolumn {%
  \global \@colroom \@colht
  \@tryfcolumn \@deferlist
  \if@fcolmade
⟨*trace⟩
    \tr@ce{PAGE: float \if@twocolumn column \else page \fi
                completed}%
⟨/trace⟩
  \else

    \begingroup
      \let \reserved@b \@deferlist
      \global \let \@deferlist \@empty
      \let \@elt \@scolelt
      \reserved@b
    \endgroup
  \fi
}
```

This one does not need to set `\@colht`.

```
\def \@startdblcolumn {%
```

Not needed since this always comes after `\@outputpage`:

```
% \global \@colht \textheight
  \@tryfcolumn \@dbldeferlist
  \if@fcolmade
⟨*trace⟩
    \tr@ce{PAGE: double float page completed}%
⟨/trace⟩
  \else

    \begingroup
      \let \reserved@b \@dbldeferlist
      \global \let \@dbldeferlist \@empty
      \let \@elt \@sdblcolelt
      \reserved@b
    \endgroup
  \fi
}
```

`\@tryfcolumn` Now tests if its list is empty before any further exertion.

```
\def \@tryfcolumn #1{%
  \global \@fcolmadefalse
  \ifx #1\@empty
  \else
⟨*trace⟩
     \tr@ce{PAGE: try float \if@twocolumn column/page\else page\fi
                  ---\string #1}%
     \tr@ce{----- \string #1: #1}%
⟨/trace⟩

    \xdef\@trylist{#1}%
    \global \let \@failedlist \@empty
    \begingroup
      \let \@elt \@xtryfc \@trylist
    \endgroup
    \if@fcolmade
      \@vtryfc #1%
    \fi
  \fi
}
⟨/2ekernel | autoload | fltrace⟩

⟨*2ekernel | autoload⟩
```

\@scolelt

```
\def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}
```

\@sdblcolelt

```
\def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}
```

\@vtryfc

```
\def\@vtryfc #1{%
  \global\setbox\@outputbox\vbox{}%
  \let\@elt\@wtryfc
  \@flsucceed
  \global\setbox\@outputbox \vbox to\@colht{%
    \vskip \@fptop
    \vskip -\@fpsep
    \unvbox \@outputbox
    \vskip \@fpbot}%
  \let\@elt\relax
  \xdef #1{\@failedlist\@flfail}%
  \xdef\@freelist{\@freelist\@flsucceed}}
```

\@wtryfc

```
\def\@wtryfc #1{%
  \global\setbox\@outputbox\vbox{%
    \unvbox\@outputbox
    \vskip\@fpsep
    \box #1}}
```

\@xtryfc

```
\def\@xtryfc #1{%
  \@next\reserved@a\@trylist{}{}%
  \@currtype \count #1%
  \divide\@currtype\@xxxii
  \multiply\@currtype\@xxxii
  \@bitor \@currtype \@failedlist
  \@testfp #1%
  \ifdim \ht #1>\@colht
    \@testtrue
  \fi
  \if@test
    \@cons\@failedlist #1%
  \else
    \@ytryfc #1%
  \fi}
```

\@ytryfc

```
\def\@ytryfc #1{%
  \begingroup
    \gdef\@flsucceed{\@elt #1}%
    \global\let\@flfail\@empty
    \@tempdima\ht #1%
    \let\@elt\@ztryfc
    \@trylist
    \ifdim \@tempdima >\@fpmin
      \global\@fcolmadetrue
    \else
      \@cons\@failedlist #1%
    \fi
  \endgroup
  \if@fcolmade
    \let\@elt\@gobble
  \fi}
```

`\@ztryfc`

```
\def\@ztryfc #1{%
  \@tempcnta \count#1%
  \divide\@tempcnta\@xxxii
  \multiply\@tempcnta\@xxxii
  \@bitor \@tempcnta {\@failedlist \@flfail}%
  \@testfp #1%
  \@tempdimb\@tempdima
  \advance\@tempdimb \ht#1%
  \advance\@tempdimb\@fpsep
  \ifdim \@tempdimb >\@colht
    \@testtrue
  \fi
  \if@test
    \@cons\@flfail #1%
  \else
    \@cons\@flsucceed #1%
    \@tempdima\@tempdimb
  \fi}
```

```
⟨/2ekernel | autoload⟩
```

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.

```
⟨*2ekernel | autoload | fltrace⟩
\def \@addtobot {%
⟨*trace⟩
   \tr@ce{***Start addtobot}%
⟨/trace⟩
   \@getfpsbit 4\relax
⟨*trace⟩
   \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi bot:
                                                  \the \@fpstype}%
⟨/trace⟩
   \ifodd \@tempcnta
     \@flsetnum \@botnum
     \ifnum \@botnum>\z@
       \@tempswafalse
       \@flcheckspace \@botroom \@botlist
       \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?

```
         \global \maxdepth \z@
         \@flupdates \@botnum \@botroom \@botlist
⟨*trace⟩
         \tr@ce{colroom (after-bot) = \the \@colroom}%
         \tr@ce{colnum (after-bot) = \the \@colnum}%
         \tr@ce{botnum (after-bot) = \the \@botnum}%
         \tr@ce{***Success: bot}%
⟨/trace⟩
         \@inserttrue
       \fi
⟨*trace⟩
     \else
       \tr@ce{Fail: botnum = \the \@botnum:
                              fpstype \the \@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \tr@ce{ERROR: !b float not successful (addtobot)}%
       \fi
⟨/trace⟩
```

```
     \fi
   \fi
}
```

`\@addtotoporbot` Lots of changes.

```
\def \@addtotoporbot {%
⟨*trace⟩
   \tr@ce{***Start addtotoporbot}%
⟨/trace⟩
   \@getfpsbit \tw@
⟨*trace⟩
   \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi top:
                                                      \the \@fpstype}%
⟨/trace⟩
   \ifodd \@tempcnta
     \@flsetnum \@topnum
     \ifnum \@topnum>\z@
       \@tempswafalse
       \@flcheckspace \@toproom \@toplist
       \if@tempswa
         \@bitor\@currtype{\@midlist\@botlist}%
⟨*trace⟩
           \tr@ce{(mid+bot)list: \@midlist, \@botlist:
                               (addtotoporbot-before)}%
⟨/trace⟩
         \if@test
⟨*trace⟩
           \tr@ce{type already on list: mid or bot---sent to addtobot}%
⟨/trace⟩
         \else
          \@flupdates \@topnum \@toproom \@toplist
⟨*trace⟩
          \tr@ce{colroom (after-top) = \the \@colroom}%
          \tr@ce{colnum (after-top) = \the \@colnum}%
          \tr@ce{topnum (after-top) = \the \@topnum}%
          \tr@ce{***Success: top}%
⟨/trace⟩
          \@inserttrue
         \fi
       \fi
⟨*trace⟩
     \else
       \tr@ce{Fail: topnum = \the \@topnum: fpstype
                                             \the \@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \tr@ce{ERROR: !t float not successful (addtotoporbot)}%
       \fi
⟨/trace⟩
     \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
     \tr@ce{sent to addtobot (addtotoporbot)}%
⟨/trace⟩
     \@addtobot
   \fi
}
⟨/2ekernel | autoload | fltrace⟩
```

`\@addtocurcol` Lots of changes.

```
⟨*2ekernel | autoload | fltrace | flafter⟩
```

```
\def \@addtocurcol {%
⟨*trace⟩
  \tr@ce{***Start addtocurcol}%
⟨/trace⟩
   \@insertfalse
   \@setfloattypecounts
   \ifnum \@fpstype=8
⟨*trace⟩
     \tr@ce{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
⟨/trace⟩
   \else
     \ifnum \@fpstype=24
⟨*trace⟩
       \tr@ce{fpstype p only (addtocurcol): \the \@fpstype = 24?}%
⟨/trace⟩
     \else
       \@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.

```
⟨*trace⟩
       \tr@ce{textfloatsheight (before) = \the \@textfloatsheight}%
⟨/trace⟩
       \advance \@textmin \@textfloatsheight
       \@reqcolroom \@pageht
```

This line must be removed since `\@specialoutput` changed.

```
%        \advance \@reqcolroom \@pagedp
⟨*trace⟩
       \tr@ce{textmin + textfloatsheight: \the \@textmin}%
       \tr@ce{page-so-far: \the \@reqcolroom}%
⟨/trace⟩
       \ifdim \@textmin>\@reqcolroom
         \@reqcolroom \@textmin
⟨*trace⟩
         \tr@ce{ORD? textmin being used}%
⟨/trace⟩
       \fi
       \advance \@reqcolroom \ht\@currbox
⟨*trace⟩
       \tr@ce{float size = \the \ht \@currbox (addtocurcol)}%
       \tr@ce{colroom = \the \@colroom (addtocurcol)}%
       \tr@ce{reqcolroom = \the \@reqcolroom (addtocurcol)}%
⟨/trace⟩
       \ifdim \@colroom>\@reqcolroom
         \@flsetnum \@colnum
         \ifnum \@colnum>\z@
           \@bitor\@currtype\@deferlist
⟨*trace⟩
           \tr@ce{deferlist: \@deferlist: (addtocurcol-before)}%
⟨/trace⟩
           \if@test
⟨*trace⟩
             \tr@ce{type already on list: defer (addtocurcol)}%
⟨/trace⟩
           \else
             \@bitor\@currtype\@botlist
⟨*trace⟩
           \tr@ce{botlist: \@botlist: (addtocurcol-before)}%
⟨/trace⟩
            \if@test
```

```
⟨*trace⟩
       \tr@ce{type already on list: bot---sent to addtobot}%
⟨/trace⟩
       \@addtobot
      \else
⟨*trace⟩
       \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi
           here: \the \@fpstype}%
⟨/trace⟩
       \ifodd \count\@currbox
        \advance \@reqcolroom \intextsep
        \ifdim \@colroom>\@reqcolroom
         \global \advance \@colnum \m@ne
         \global \advance \@textfloatsheight \ht\@currbox
```

This may sometimes give an overestimate.

```
         \global \advance \@textfloatsheight 2\intextsep
         \@cons \@midlist \@currbox
⟨*trace⟩
         \tr@ce{***Success: here}%
         \tr@ce{textfloatsheight (after-here) =
              \the \@textfloatsheight}%
         \tr@ce{colnum (after-here) = \the \@colnum}%
⟨/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 betweeen paragraphs (i.e. when in vertical mode).

If the nobreak switch is true we need to reset it and clear `\everypar` sionce the float may not reset the flag and cannot reset the `\everypar` globally.

Typesetting starts here (we are in vertical mode).

```
         \if@nobreak
          \nobreak
          \@nobreakfalse
          \everypar{}%
         \else
          \addpenalty \interlinepenalty
         \fi
         \vskip \intextsep
         \box\@currbox
         \penalty\interlinepenalty
         \vskip\intextsep
         \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi
```

Typesetting ends here.

```
         \outputpenalty \z@
         \@inserttrue
⟨*trace⟩
        \else
         \tr@ce{Fail---no room at 2nd test of colroom
                 (addtocorcol \string\intextsep)}%
⟨/trace⟩
        \fi
       \fi
       \if@insert
       \else
```

```
⟨*2ekernel | autoload | fltrace⟩
⟨*trace⟩
                 \tr@ce{not here: sent to addtotoporbot}%
⟨/trace⟩
                 \@addtotoporbot
⟨/2ekernel | autoload | fltrace⟩
⟨*!2ekernel&!autoload&!fltrace⟩
⟨*trace⟩
                 \tr@ce{not here: sent to addtobot}%
⟨/trace⟩
                 \@addtobot
⟨/!2ekernel&!autoload&!fltrace⟩
               \fi
             \fi
           \fi
⟨*trace⟩
         \else
           \tr@ce{Fail: colnum = \the \@colnum:
                        fpstype \the \@fpstype=ORD?}%
           \ifnum \@fpstype<\sixt@@n
             \tr@ce{ERROR: BANG float not successful (addtocurcol)}%
           \fi
⟨/trace⟩
         \fi
⟨*trace⟩
       \else
         \tr@ce{Fail---no room: fl box ht: \the \ht \@currbox
                                                     (addtocurcol)}%
⟨/trace⟩
       \fi
     \fi
   \fi
   \if@insert
   \else
     \@resethfps
⟨*trace⟩
     \tr@ce{put on deferlist (addtocurcol)}%
⟨/trace⟩
     \@cons\@deferlist\@currbox
⟨*trace⟩
     \tr@ce{deferlist: \@deferlist: (addtocurcol-after)}%
⟨/trace⟩
   \fi
}
⟨/2ekernel | autoload | fltrace | flafter⟩
```

`\@addtonextcol` Lots of changes.

```
⟨*2ekernel | autoload | fltrace⟩
\def\@addtonextcol{%
  \begingroup
⟨*trace⟩
   \tr@ce{***Start addtonextcol}%
⟨/trace⟩
   \@insertfalse
   \@setfloattypecounts
   \ifnum \@fpstype=8
⟨*trace⟩
     \tr@ce{fpstype not curcol: \the \@fpstype = 8?}%
⟨/trace⟩
   \else
     \ifnum \@fpstype=24
⟨*trace⟩
```

```
       \tr@ce{fpstype not curcol: \the \@fpstype = 24?}%
⟨/trace⟩
     \else
       \@flsettextmin
⟨*trace⟩
       \tr@ce{text-so-far: 0pt (top of col)}%
⟨/trace⟩
       \@reqcolroom \ht\@currbox
⟨*trace⟩
       \tr@ce{float size: \the \@reqcolroom (addtonextcol)}%
⟨/trace⟩
       \advance \@reqcolroom \@textmin
⟨*trace⟩
       \tr@ce{colroom = \the \@colroom (addtonextcol)}%
       \tr@ce{reqcolroom = \the \@reqcolroom (addtonextcol)}%
⟨/trace⟩
       \ifdim \@colroom>\@reqcolroom
         \@flsetnum \@colnum
         \ifnum\@colnum>\z@
            \@bitor\@currtype\@deferlist
⟨*trace⟩
            \tr@ce{deferlist: \@deferlist: (addtonextcol-before)}%
⟨/trace⟩
            \if@test
⟨*trace⟩
              \tr@ce{type already on list: defer (addtonextcol)}%
⟨/trace⟩
            \else
⟨*trace⟩
              \tr@ce{sent to addtotoporbot (addtonextcol)}%
⟨/trace⟩
              \@addtotoporbot
            \fi
         \fi
⟨*trace⟩
       \else
         \tr@ce{Fail---no room: fl box ht: \the \ht \@currbox
                                                  (addtonextcol)}%
⟨/trace⟩
       \fi
     \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
     \tr@ce{put back on deferlist (addtonextcol)}%
⟨/trace⟩
     \@cons\@deferlist\@currbox
⟨*trace⟩
     \tr@ce{deferlist: \@deferlist: (addtonextcol-after)}%
⟨/trace⟩
   \fi
⟨*trace⟩
   \tr@ce{End of addtonextcol -- locally counts:}%
   \tr@ce{ col: \the \@colnum. top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
  \endgroup
⟨*trace⟩
  \tr@ce{End of addtonextcol -- globally counts:}%
  \tr@ce{col: \the \@colnum. top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
}
```

`\@addtodblcol` Lots of changes.

```
\def\@addtodblcol{%
  \begingroup
⟨*trace⟩
  \tr@ce{***Start addtodblcol}%
⟨/trace⟩
   \@insertfalse
   \@setfloattypecounts
   \@getfpsbit \tw@
⟨*trace⟩
   \tr@ce{fpstype \ifodd \@tempcnta OK \else not \fi dbltop:
                                                     \the \@fpstype}%
⟨/trace⟩
   \ifodd\@tempcnta
     \@flsetnum \@dbltopnum
     \ifnum \@dbltopnum>\z@
       \@tempswafalse
       \ifdim \@dbltoproom>\ht\@currbox
         \@tempswatrue
⟨*trace⟩
         \tr@ce{Space OK: \@dbltoproom =
                \the \@dbltoproom > \the \ht \@currbox
                                         (dbltoproom)}%
⟨/trace⟩
       \else
⟨*trace⟩
         \tr@ce{fpstype: \the \@fpstype (addtodblcol)}%
⟨/trace⟩
         \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
           \tr@ce{BANG float ignoring \@dbltoproom}%
           \tr@ce{\@spaces \@dbltoproom = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox-BANG}%
⟨/trace⟩
```

Need to check that there is room on the page, using the local value of `\@textmin` to make the necessary adjustment to `\@dbltoproom`.

```
           \advance \@dbltoproom \@textmin
⟨*trace⟩
           \tr@ce{Local value of texmin: \the\@textmin}%
           \tr@ce{\@spaces space on page = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox-BANG}%
⟨/trace⟩
           \ifdim \@dbltoproom>\ht\@currbox
             \@tempswatrue
⟨*trace⟩
             \tr@ce{Space OK BANG: space on page = \the \@dbltoproom >
                                              \the \ht \@currbox}%
           \else
             \tr@ce{fpstype: \the \@fpstype}%
             \tr@ce{Fail---no room dbltoproom-BANG?:}%
             \tr@ce{\@spaces space on page = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox}%
⟨/trace⟩
           \fi
           \advance \@dbltoproom -\@textmin
⟨*trace⟩
         \else
           \tr@ce{fpstype: \the \@fpstype}%
           \tr@ce{Fail---no room dbltoproom-ORD?:}%
           \tr@ce{\@spaces \@dbltoproom = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox}%
```

```
⟨/trace⟩
         \fi
       \fi
       \if@tempswa
           \@bitor \@currtype \@dbldeferlist
⟨*trace⟩
           \tr@ce{dbldeferlist: \@dbldeferlist: (before)}%
⟨/trace⟩
           \if@test
⟨*trace⟩
              \tr@ce{type already on list: dbldefer}%
⟨/trace⟩
           \else
              \@tempdima -\ht\@currbox
              \advance\@tempdima
                -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
                                          \dblfloatsep \fi
              \global \advance \@dbltoproom \@tempdima
              \global \advance \@colht \@tempdima
              \global \advance \@dbltopnum \m@ne
              \@cons \@dbltoplist \@currbox
⟨*trace⟩
              \tr@ce{dbltopnum (after) = \the \@dbltopnum}%
              \tr@ce{***Success: dbltop}%
⟨/trace⟩
              \@inserttrue
           \fi
       \fi
⟨*trace⟩
     \else
       \tr@ce{Fail: dbltopnum = \the \@dbltopnum: fpstype
                                                  \the \@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \tr@ce{ERROR: !t float not successful (addtodblcol)}%
       \fi
⟨/trace⟩
     \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
     \tr@ce{put on dbldeferlist}%
⟨/trace⟩
     \@cons\@dbldeferlist\@currbox
⟨*trace⟩
     \tr@ce{dbldeferlist: \@dbldeferlist: (after)}%
⟨/trace⟩
   \fi
⟨*trace⟩
   \tr@ce{End of addtodblcol -- locally count:}%
   \tr@ce{ dbltop: \the \@dbltopnum.}%
⟨/trace⟩
  \endgroup
⟨*trace⟩
  \tr@ce{End of addtodblcol -- globally count:}%
  \tr@ce{dbltop: \the \@dbltopnum.}%
⟨/trace⟩
}
⟨/2ekernel | autoload | fltrace⟩
```

`\@addmarginpar`

```
⟨*2ekernel | autoload⟩
\def\@addmarginpar{\@next\@marbox\@currlist{\@cons\@freelist\@marbox
\@cons\@freelist\@currbox}\@latexbug\@tempcnta\@ne
\if@twocolumn
    \if@firstcolumn \@tempcnta\m@ne \fi
\else
  \if@mparswitch
     \ifodd\c@page \else\@tempcnta\m@ne \fi
  \fi
  \if@reversemargin \@tempcnta -\@tempcnta \fi
\fi
\ifnum\@tempcnta <\z@  \global\setbox\@marbox\box\@currbox \fi
\@tempdima\@mparbottom
\advance\@tempdima -\@pageht
\advance\@tempdima\ht\@marbox
\ifdim\@tempdima >\z@
  \@latex@warning@no@line {Marginpar on page \thepage\space moved}%
\else
  \@tempdima\z@
\fi
\global\@mparbottom\@pageht
\global\advance\@mparbottom\@tempdima
\global\advance\@mparbottom\dp\@marbox
\global\advance\@mparbottom\marginparpush
\advance\@tempdima -\ht\@marbox
```

Putting box movement inside the ‘marbox’:

```
\global\setbox \@marbox
               \vbox {\vskip \@tempdima
                      \box \@marbox}%
\global \ht\@marbox \z@
\global \dp\@marbox \z@
```

Sticking (rather than gluing:-) the ‘marbox’ to the line above, changed vskip to kern:

```
\kern -\@pagedp
\nointerlineskip
\hb@xt@\columnwidth
  {\ifnum \@tempcnta >\z@
      \hskip\columnwidth \hskip\marginparsep
   \else
      \hskip -\marginparsep \hskip -\marginparwidth
   \fi
   \box\@marbox \hss}%
```

For this reason the following code can vanish:

```
\nobreak             %% No longer needed.  CAR92/12
\vskip -\@tempdima   %% No longer needed.  CAR92/12
```

```
\nointerlineskip
\hbox{\vrule \@height\z@ \@width\z@ \@depth\@pagedp}}
```

⟨/2ekernel | autoload⟩

66.1.1 Kludgeins

This part of the file is part of the implementation of the following two new commands for LaTeX2e.

```
\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 execpt 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 TEX do a lot of the necessary work. All we need to put into it is the amount by which the pagegoal should be changed.

```
⟨*2ekernel | def1⟩
\newinsert \@kludgeins
\global\dimen\@kludgeins \maxdimen
\global\count\@kludgeins 1000
⟨/2ekernel | def1⟩
```

`\enlargethispage` The user command.

`\enlargethispage*`

```
⟨*2ekernel | def1⟩
\gdef \enlargethispage {%
   \@ifstar
     {%
⟨*trace⟩
      \tr@ce{Enlarging page height * }%
⟨/trace⟩
      \@enlargepage{\hbox{\kern\p@}}}%
     {%
⟨*trace⟩
      \tr@ce{Enlarging page height exactly---}%
⟨/trace⟩
      \@enlargepage\@empty}%
}
⟨/2ekernel | def1⟩
⟨*autoload⟩
\def\enlargethispage{\@autoload{out1}\enlargethispage}
⟨/autoload⟩
```

`\@enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```
⟨*2ekernel | def1⟩
\gdef\@enlargepage#1#2{%
⟨*trace⟩
   \tr@ce{\@spaces\@spaces by #2}%
⟨/trace⟩
   \@tempskipa#2\relax
   \ifdim \@tempskipa>.5\maxdimen
     \@latexerr{Suggested\space extra\space height\space
                (\the\@tempskipa)\space dangerously\space
                large}\@eha
   \else
     \ifdim \vsize<.5\maxdimen
⟨*trace⟩
       \tr@ce {Kludgeins added--pagegoal before: \the\pagegoal}%
⟨/trace⟩
       \@bsphack
         \insert\@kludgeins{#1\vskip-\@tempskipa}%
       \@esphack
```

This next bit is for tracing only:

```
⟨*trace⟩
       \ifvmode \par
         \tr@ce {Kludgeins added--pagegoal after: \the \pagegoal}%
       \fi
⟨/trace⟩
     \else
       \@latexerr{Page\space height\space already\space
                  too\space large}\@eha
     \fi
   \fi
}
⟨/2ekernel | def1⟩
```

66.1.2 Float control

This part implements controllable floats and other changes to the float mechanism.

It provides, at the doument level, the following command for inclusion in LaTeX2e.

`\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 followinhg 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 suplies a reasonably exhaustive, and somewhat baroque, means of tracing some aspects of the float mechanism.

More tracing.

`\tr@ce`
`\notrace`
`\tracefloats`
`\@traceval`
`\tracefloatvals`
`\@tracemessage`

Set-up tracing for floats independent of other tracing as it produces mega-output. Default is no tracing.

```
⟨*trace⟩
\def \@tracemessage #1{\typeout{LaTeX2e: #1}}
\def \tracefloats{\let \tr@ce \@tracemessage}
\def \notrace {\let \tr@ce \@gobble}
\notrace
\def \@traceval #1{\tr@ce{\string #1 = \the #1}}
\def \tracefloatvals{%
  \@dblfloatplacement
  \@floatplacement
  \@traceval\@colnum
  \@traceval\@colroom
  \@traceval\@topnum
  \@traceval\@toproom
  \@traceval\@botnum
  \@traceval\@botroom
  \@traceval\@fpmin
  \tr@ce{\string\textfraction = \textfraction}%
  \@traceval\@dbltopnum
  \@traceval\@dbltoproom
}
⟨/trace⟩
⟨*flafter⟩
\providecommand\tr@ce[1]{}
⟨/flafter⟩
```

`\suppressfloats` `\@flstop` Float suppression commands: these set the relevant counter globally to zero. Thus they are overridden for a particular float by an ! specifier.

```
⟨*2ekernel | autoload⟩
\def \suppressfloats {%
  \@ifnextchar [%
    \@flstop
   {\global \@colnum \z@}%
}
```

Maybe this should be a loop over `#1`?

```
\def \@flstop [#1]{%
  \if t#1%
    \global \@topnum \z@
  \fi
  \if b#1%
    \global \@botnum \z@
  \fi
}
```

Manipulation of float placement and type; both their strings and the corresponding count registers.

`\@fpstype` `\@reqcolroom` `\@textfloatsheight` First a new count register to go with `\@currtype`.

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.

```
\newcount \@fpstype
\newdimen \@reqcolroom
\newdimen \@textfloatsheight
⟨/2ekernel | autoload⟩
```

`\@fpsadddefault` Adds the default placement to what is already there.

Should not need to change this, but could do it as follows:

```
\def \@fpsadddefault {%
   \@temptokena \expandafter\expandafter\expandafter
                {\csname fps@\@captype \endcsname}%
   \edef \reserved@a {\the\@temptokena}%
   \@onelevel@sanitize \reserved@a
   \edef \@fps {\@fps\reserved@a}%
}
```

```
⟨*2ekernel | autoload | fltrace⟩
\def \@fpsadddefault {%
⟨*trace⟩
  \tr@ce{fps changed from: \@fps}%
⟨/trace⟩
  \edef \@fps {\@fps\csname fps@\@captype \endcsname}%
  \@latex@warning {%
    No positions in optional float specifier.\MessageBreak
    Default added (so using `\@fps')}%
}
```

`\@setfloattypecounts` Sets counters `\@fpstype` and `\@currtype`.

BANG == bit4 of `\count\@currbox` = 0.

```
\def \@setfloattypecounts {%
  \@currtype \count\@currbox
  \@fpstype \count\@currbox
  \divide\@currtype\@xxxii \multiply\@currtype\@xxxii
  \advance \@fpstype -\@currtype
⟨*trace⟩
  \tr@ce{(mod 32) fpstype: \the \@fpstype}%
  \tr@ce{(mult of 32) currtype: \the \@currtype}%
% Tracing only: but some should be changed into real errors/warnings?
  \ifnum \@fpstype<\sixt@@n
    \ifnum \@fpstype=\z@
      \tr@ce{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
    \fi
    \ifnum \@fpstype=\@ne
      \tr@ce{WARNING: only h, fpstype = \the \@fpstype = 1?}%
    \fi
    \tr@ce{BANG float}%
  \else
    \ifnum \@fpstype=\sixt@@n
      \tr@ce{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
    \fi
    \ifnum \@fpstype=17
      \tr@ce{WARNING: only h, fpstype = \the \@fpstype = 17?}%
    \fi
    \tr@ce{ORD float}%
  \fi
⟨/trace⟩
}
⟨/2ekernel | autoload | fltrace⟩
```

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```
⟨*2ekernel | autoload⟩
\def \@getfpsbit {%
   \@boxfpsbit \@currbox
}
```

`\@boxfpsbit` Used above.

```
\def \@boxfpsbit #1#2{%
   \@tempcnta \count#1%
   \divide \@tempcnta #2\relax
}
```

`\@testfp` New definition of the float page test.

```
\def \@testfp #1{%
   \@boxfpsbit #18\relax % Really ‘#1 8’ for human readers!
   \ifodd \@tempcnta
   \else
     \@testtrue
   \fi
}
```

`\@setfpsbit` Sets required bit of `\@tempcnta` (to 1).

```
\def \@setfpsbit #1{%
   \@tempcntb \@tempcnta
   \divide \@tempcntb #1\relax
   \ifodd \@tempcntb
   \else
```

```
    \advance \@tempcnta #1\relax
  \fi
}
⟨/2ekernel | autoload⟩
```

`\@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`.

```
⟨*2ekernel | autoload | fltrace⟩
\def \@resethfps {%
  \let\reserved@a\@empty
  \ifnum \@fpstype=\@ne
     \def \reserved@a {!}%
     \@fpstype 17
  \fi
  \ifnum \@fpstype=17
    \global \advance \count\@currbox \tw@
    \@latex@warning@no@line {%
      ‘\reserved@a h’ float specifier changed to ‘\reserved@a ht’}%
⟨*trace⟩
    \tr@ce{%
       ‘t’ added to ‘\reserved@a h’- new Count: \the \count\@currbox}%
⟨/trace⟩
  \fi
}
```

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 assgnments 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.

```
\def \@flsetnum #1{%
⟨*trace⟩
  \tr@ce{fpstype: \the \@fpstype (flsetnum \string#1)}%
⟨/trace⟩
  \ifnum \@fpstype<\sixt@@n
    \ifnum #1=\z@
⟨*trace⟩
      \tr@ce{BANG float resetting \string#1 to 1}%
⟨/trace⟩
      #1\@ne
    \fi
  \fi
⟨*trace⟩
  \tr@ce{#1 (before) = \the #1}%
⟨/trace⟩
}
```

`\@flsettextmin` This ignores `\textfraction` space restriction in case BANG.

```
\def \@flsettextmin {%
⟨*trace⟩
  \tr@ce{fpstype: \the \@fpstype (flsettextmin)}%
⟨/trace⟩
   \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
     \tr@ce{BANG ignoring textmin}%
⟨/trace⟩
     \@textmin \z@
   \else
     \@textmin \textfraction\@colht
⟨*trace⟩
     \tr@ce{ORD textmin = \the \@textmin}%
⟨/trace⟩
   \fi
}
```

`\@flcheckspace` This ignores space restriction in case BANG; this is still slightly conervative since it does not allow for the fact that, if there is no text in the column then `\textfloatsep` is not needed. Sets `@tempswa` true if there is room for `\@currbox`.

```
\def \@flcheckspace #1#2{%
   \advance \@reqcolroom
     \ifx #2\@empty \textfloatsep \else \floatsep \fi
⟨*trace⟩
   \tr@ce{colroom = \the \@colroom (flcheckspace \string#1 \string#2)}%
   \tr@ce{reqcolroom = \the \@reqcolroom
                                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
   \ifdim \@colroom>\@reqcolroom
     \ifdim #1>\ht\@currbox
       \@tempswatrue
⟨*trace⟩
       \tr@ce{Space OK: #1 = \the #1 > \the \ht \@currbox
                                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
     \else
⟨*trace⟩
       \tr@ce{fpstype: \the \@fpstype
                                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
       \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
         \tr@ce{BANG float ignoring #1
                                   (flcheckspace \string#1 \string#2):}%
         \tr@ce{\@spaces #1 = \the #1.  Ht float: \the \ht \@currbox
                                                          BANG}%
⟨/trace⟩
         \@tempswatrue
⟨*trace⟩
       \else
         \tr@ce{Fail---no room (flcheckspace \string#1 \string#2)
                       (fpstype \the \@fpstype=ORD?):}%
         \tr@ce{\@spaces #1 = \the #1.  Ht float: \the \ht \@currbox
                                                          ORD?}%
⟨/trace⟩
       \fi
     \fi
⟨*trace⟩
   \else
     \tr@ce{Fail---no room at 2nd test of colroom
                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
   \fi
}
⟨/2ekernel | autoload | fltrace⟩
```

`\@flupdates` This updates everything when a float is placed.

```
⟨*2ekernel | autoload⟩
\def \@flupdates #1#2#3{%
   \global \advance #1\m@ne
   \global \advance \@colnum \m@ne
   \@tempdima -\ht\@currbox
   \advance \@tempdima
     -\ifx #3\@empty \textfloatsep \else \floatsep \fi
   \global \advance #2\@tempdima
   \global \advance \@colroom \@tempdima
   \@cons #3\@currbox
}
⟨/2ekernel | autoload⟩
```

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 `\dbltexfloatsep`'. 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 `\intextsep`: 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 immediaiely follows other spaces. This is also possibly too big a change, at least for compatibility mode? Or it may be simply wrong! It has not been changed.
9. Now `\@addtocurcol` checks first for just p fps. I think that this is an increase in efficiency, but maybe the coding should be made even more efficient.
10. `\@tryfcolumn` now tests if the list is empty first, otherwise lots of wasted time! Thus this test has been removed from `\@startcolumn`. As Frank pointed out, this makes `\@startcolumn` less efficient. But it is now the same as `\@startdblcolumn`: I can see no reason why they should be different, but which is best?
11. Why is `\@colroom` set in `\@doclearpage`?
12. Footnotes. Check what `\clearpage` does when footnotes are left over. Footnotes are not put on float pages and, also, `\@addtonextcol` ignores the existence of held-over footnotes in deciding what floats can go on the page. Not changed.

13. `\clearpage` can still lose non-boxes, at least when floats are involved. It also moves some to the 'wrong page', but this may be a coding problem.

14. The ! option makes it necessary to check in `\output` that there is enough room left on the page after adding a float. (This would have been necessary anyway if anyone set `\@textmin` too close to zero! A similar danger existed also if the text in a `\twocolumn[text]` entity gets too large.) The current implementation of this also makes the normal case a little less efficient, OK? Not enough room means, at present, less than `\baselineskip`, with a warning: is this OK? Should it be made generic (another parameter)?

15. There are four possibilities for supporting this:

    `\twocolumn[\maketitle more text]`

    One is to change `\maketitle` slightly to allow this. Another is to change `\@topnewpage` so that more than one `\twocolumn[]` command is allowed; in this case `\maketitle\twoclumn[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 beeen done.

16. Where should the reinserts be put to maximise the probability that footmotes 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 adjustement 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.

```
⟨*2ekernel | autoload | fltrace⟩
\def\@makefcolumn #1{%
  \begingroup
    \@fpmin \z@
    \let \@testfp \@gobble
    \@tryfcolumn #1%
  \endgroup
⟨*trace⟩
  \if@fcolmade
    \tr@ce{PAGE: in \string\clearpage \if@twocolumn ---twocolumn\fi---}%
    \tr@ce{----- float column/page completed from \string#1}%
  \fi
⟨/trace⟩
}
```

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.

```
\def\@outputdblcol{%
  \if@firstcolumn
    \global \@firstcolumnfalse
    \global \setbox\@leftcolumn \box\@outputbox
⟨*trace⟩
    \tr@ce{PAGE: first column boxed}%
⟨/trace⟩
  \else
    \global \@firstcolumntrue
    \setbox\@outputbox \vbox {%
                         \hb@xt@\textwidth {%
                           \hb@xt@\columnwidth {%
                             \box\@leftcolumn \hss}%
                           \hfil
                           {\normalcolor\vrule \@width\columnseprule}%
                           \hfil
                           \hb@xt@\columnwidth {%
                             \box\@outputbox \hss}%
                                             }%
                              }%
⟨*trace⟩
    \tr@ce{PAGE: second column also boxed}%
⟨/trace⟩
    \@combinedblfloats
    \@outputpage
⟨*trace⟩
    \tr@ce{PAGE: two column page completed}%
⟨/trace⟩
    \begingroup
      \@dblfloatplacement
      \@startdblcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startdblcolumn`.

```
      \@whilesw\if@fcolmade \fi
        {\@outputpage
⟨*trace⟩
      \tr@ce{PAGE: double float page completed}%
⟨/trace⟩
         \@startdblcolumn}%
    \endgroup
  \fi
}
⟨/2ekernel | autoload | fltrace⟩
```

66.1.3 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 douments 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.

```
⟨*2ekernel | autoload⟩
\newcount\c@topnumber
\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.

```
\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.

```
\newcount\c@bottomnumber
\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.

```
\newcommand\bottomfraction{.3}
```

`\c@totalnumber` This counter holds the maximum number of floats that can appear on any text page or column.

```
\newcount\c@totalnumber
\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.

```
\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.

```
\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.

```
\newcount\c@dbltopnumber
\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.

```
\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.

```
\newcommand\dblfloatpagefraction{.5}
```

Floats on a text page

`\floatsep` `\textfloatsep` `\intextsep` When a floating object is placed on a page with text, these parameters control the seperation between the float and the other objects on the page. These parameters are used for both one-column mode and single-column floats in two-column mode. They are all rubber lengths.

`\floatsep` is the space between adjacent floats that are placed at the top or bottom of the text page or column.

`\textfloatsep` is the space between the main text and floats at the top or bottom of the page or column.

`\intextsep` is the space between in-text floats and the text.

```
\newskip\floatsep
\newskip\textfloatsep
\newskip\intextsep
\setlength\floatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\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.

```
\newskip\dblfloatsep
\newskip\dbltextfloatsep
\setlength\dblfloatsep    {12\p@ \@plus 2\p@ \@minus 2\p@}
\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 seperate 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` ais 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.

```
\newskip\@fptop
\newskip\@fpsep
\newskip\@fpbot
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
```

`\@dblfptop`
`\@dblfpsep`
`\@dblfpbot`

Double-column 'float pages' in two-column mode use similar parameters.

```
\newskip\@dblfptop
\newskip\@dblfpsep
\newskip\@dblfpbot
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
```

`\topfigrule`
`\botfigrule`
`\dblfigrule`

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.

```
\let\topfigrule=\relax
\let\botfigrule=\relax
\let\dblfigrule=\relax
⟨/2ekernel | autoload⟩
```

File L

ltclass.dtx

67 Introduction

This file implements the following declarations, which replace `\documentstyle` in LaTeX 2ε documents.

Note that old documents containing `\documentstyle` will be run using a compatibility option—thus keeping everyone happy, we hope!

The overall idea is that there are two types of 'style files': 'class files' which define elements and provide a default formatting for them; and 'packages' which provide extra functionality. One difference between LaTeX 2ε and LaTeX2.09 is that LaTeX 2ε 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.

68 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 LaTeX2.09 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 LaTeX2.09 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 ⟨*package-option-list*⟩, each package processes the ⟨*main-option-list*⟩. 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.

68.1 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.

69 Class and Package interface

69.1 Class name and version

`\ProvidesClass` A class can identify itself with the `\ProvidesClass{`⟨*name*⟩`}[`⟨*version*⟩`]` command. The ⟨*version*⟩ should begin with a date in the format `YYYY/MM/DD`.

69.2 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`.

69.3 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`
`\@ifclassloaded`
`\@ifpackagelater`
`\@ifclasslater`
`\@ifpackagewith`
`\@ifclasswith`

To find out if a package has already been loaded, use `\@ifpackageloaded{`⟨*package*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

To find out if a package has already been loaded with a version more recent than ⟨*version*⟩, use `\@ifpackagelater{`⟨*package*⟩`}{`⟨*version*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

To find out if a package has already been loaded with at least the options ⟨*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).

69.4 Declaring new options

Options for classes and packages are built using the same macros.

`\DeclareOption`
`\DeclareOption*`

To define a builtin option, use `\DeclareOption{`⟨*name*⟩`}{`⟨*code*⟩`}`.

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*{\@unkownoptionerror}`

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.

69.5 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 LaTeX2.09 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.

70 Implementation

```
⟨*2ekernel⟩
```

`\if@compatibility`

The flag for compatibilty mode.

```
\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`.

```
\def\@documentclasshook{%
  \ifx\@normalsize\@undefined
     \let\@normalsize\normalsize
  \fi
}
```

`\@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`.

```
\let\@declaredoptions\@empty
```

`\@classoptionslist` List of options of the main class.

```
\let\@classoptionslist\relax
\@onlypreamble\@classoptionslist
```

`\@unusedoptionlist` List of options of the main class that haven't been declared or loaded as class option files.

```
\let\@unusedoptionlist\@empty
\@onlypreamble\@unusedoptionlist
```

`\CurrentOption` Name of current package or option.

```
\let\CurrentOption\@empty
```

`\@currname` Name of current package or option.

```
\let\@currname\@empty
```

`\@currext` The current file extension.

```
\global\let\@currext=\@empty
```

`\@clsextension`
`\@pkgextension`
The two possible values of `\@currext`.

```
\def\@clsextension{cls}
\def\@pkgextension{sty}
\@onlypreamble\@clsextension
\@onlypreamble\@pkgextension
```

`\@pushfilename`
`\@popfilename`
`\@currnamestack`
Commands to push and pop the file name and extension.
`#1` current name.
`#2` current extension.
`#3` current catcode of `@`.
`#4` Rest of the stack.

```
\def\@pushfilename{%
  \xdef\@currnamestack{%
    {\@currname}%
    {\@currext}%
    {\the\catcode`\@}%
    \@currnamestack}}
\@onlypreamble\@pushfilename
```

```
\def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil}
\@onlypreamble\@popfilename
```

```
\def\@p@pfilename#1#2#3#4\@nil{%
  \gdef\@currname{#1}%
  \gdef\@currext{#2}%
  \catcode`\@#3\relax
  \gdef\@currnamestack{#4}}
\@onlypreamble\@p@pfilename
```

```
\gdef\@currnamestack{}
\@onlypreamble\@currnamestack
```

`\@ptionlist` Returns the option list of the file.

```
\def\@ptionlist#1{%
  \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}}
\@onlypreamble\@ptionlist
```

`\@ifpackageloaded`
`\@ifclassloaded`

`\@ifpackageloaded{`⟨*name*⟩`}` Checks to see whether a file has been loaded.

```
\def\@ifpackageloaded{\@ifl@aded\@pkgextension}
\def\@ifclassloaded{\@ifl@aded\@clsextension}
\@onlypreamble\@ifpackageloaded
\@onlypreamble\@ifclassloaded
```

```
\def\@ifl@aded#1#2{%
  \expandafter\ifx\csname ver@#2.#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}
\@onlypreamble\@ifl@aded
```

`\@ifpackagelater`
`\@ifclasslater`

`\@ifpackagelater{`⟨*name*⟩`}{YYYY/MM/DD}` Checks that the package loaded is more recent than the given date.

```
\def\@ifpackagelater{\@ifl@ter\@pkgextension}
\def\@ifclasslater{\@ifl@ter\@clsextension}
\@onlypreamble\@ifpackagelater
\@onlypreamble\@ifclasslater
```

```
\def\@ifl@ter#1#2{%
  \expandafter\@ifl@t@r
    \csname ver@#2.#1\endcsname}
\@onlypreamble\@ifl@ter
```

This internal macro is also used in `\NeedsTeXFormat`.

```
\def\@ifl@t@r#1#2{%
  \ifnum\expandafter\@parse@version#1//00\@nil<%
        \expandafter\@parse@version#2//00\@nil
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}
\@onlypreamble\@ifl@t@r
```

```
\def\@parse@version#1/#2/#3#4#5\@nil{#1#2#3#4 }
\@onlypreamble\@parse@version
```

`\@ifpackagewith`
`\@ifclasswith`

`\@ifpackagewith{`⟨*name*⟩`}{`⟨*option-list*⟩`}` Checks that ⟨*option-list*⟩ is a subset of the options **with** which ⟨*name*⟩ was loaded.

```
\def\@ifpackagewith{\@if@ptions\@pkgextension}
\def\@ifclasswith{\@if@ptions\@clsextension}
\@onlypreamble\@ifpackagewith
\@onlypreamble\@ifclasswith
```

```
\def\@if@ptions#1#2{%
  \@expandtwoargs\@if@pti@ns{\@ptionlist{#2.#1}}}
\@onlypreamble\@if@ptions
```

Probably shouldnt use `\CurrentOption` here… (changed to `\reserved@b`.)

```
\def\@if@pti@ns#1#2{%
  \let\reserved@a\@firstoftwo
  \@for\reserved@b:=#2\do{%
    \expandafter\in@\expandafter{\expandafter,\reserved@b,}{,#1,}%
    \ifin@\else\let\reserved@a\@secondoftwo\fi}%
  \reserved@a}
\@onlypreamble\@if@pti@ns
```

`\ProvidesPackage`

Checks that the current filename is correct, and defines `\ver@filename`.

```
\def\ProvidesPackage#1{%
  \xdef\@gtempa{#1}%
  \ifx\@gtempa\@currname\else
    \@latex@warning@no@line{You have requested
```

```
      \@cls@pkg\space`\@currname',\MessageBreak
       but the \@cls@pkg\space provides `#1'}%
  \fi
  \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}}%]
\@onlypreamble\ProvidesPackage
```

```
\def\@pr@videpackage[#1]{%
  \expandafter\xdef\csname ver@\@currname.\@currext\endcsname{#1}%
  \ifx\@currext\@clsextension
    \typeout{Document Class: \@gtempa\space#1}%
  \else
    \wlog{Package: \@gtempa\space#1}%
  \fi}
\@onlypreamble\@pr@videpackage
```

`\ProvidesClass` Like `\ProvidesPackage`, but for classes.

```
\let\ProvidesClass\ProvidesPackage
\@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`

```
\def\ProvidesFile#1{%
  \begingroup
    \catcode`\ 10 %
    \ifnum \endlinechar<256 %
      \ifnum \endlinechar>\m@ne
        \catcode\endlinechar 10 %
      \fi
    \fi
    \@makeother\/%
    \@makeother\&%
```

```
    \kernel@ifnextchar[{\@providesfile{#1}}{\@providesfile{#1}[]}}
```

During initex a special version of `\@providesfile` is used. The real definition is installed right at the end, in `ltfinal.dtx`.

```
\def\@providesfile#1[#2]{%
    \wlog{File: #1 #2}%
    \expandafter\xdef\csname ver@#1\endcsname{#2}%
  \endgroup}
    \end{macrocode}
```

`\PassOptionsToPackage` `\PassOptionsToClass` If the package has been loaded, we check that it was first loaded with the options. Otherwise we add the option list to that of the package.

```
\def\@pass@ptions#1#2#3{%
  \expandafter\xdef\csname opt@#3.#1\endcsname{%
    \@ifundefined{opt@#3.#1}\@empty
      {\csname opt@#3.#1\endcsname,}%
    \zap@space#2 \@empty}}
\@onlypreamble\@pass@ptions
```

```
\def\PassOptionsToPackage{\@pass@ptions\@pkgextension}
\def\PassOptionsToClass{\@pass@ptions\@clsextension}
\@onlypreamble\PassOptionsToPackage
\@onlypreamble\PassOptionsToClass
```

`\DeclareOption` Adds an option as a `\ds@` command, or the default `\default@ds` command.

`\DeclareOption*`

```
\def\DeclareOption{%
  \let\@fileswith@pti@ns\@badrequireerror
  \@ifstar\@defdefault@ds\@declareoption}
\long\def\@declareoption#1#2{%
```

```
   \xdef\@declaredoptions{\@declaredoptions,#1}%
   \toks@{#2}%
   \expandafter\edef\csname ds@#1\endcsname{\the\toks@}}
\long\def\@defdefault@ds#1{%
  \toks@{#1}%
  \edef\default@ds{\the\toks@}}
\@onlypreamble\DeclareOption
\@onlypreamble\@declareoption
\@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.

```
\def\OptionNotUsed{%
  \ifx\@currext\@clsextension
    \xdef\@unusedoptionlist{%
      \ifx\@unusedoptionlist\@empty\else\@unusedoptionlist,\fi
      \CurrentOption}%
  \fi}
\@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*`.

```
% \let\default@ds\OptionNotUsed
```

`\ProcessOptions`
`\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.

```
\def\ProcessOptions{%
  \let\ds@\@empty
  \edef\@curroptions{\@ptionlist{\@currname.\@currext}}%
  \@ifstar\@xprocess@ptions\@process@ptions}
\@onlypreamble\ProcessOptions
```

```
\def\@process@ptions{%
  \@for\CurrentOption:=\@declaredoptions\do{%
    \ifx\CurrentOption\@empty\else
      \@expandtwoargs\in@{,\CurrentOption,}{%
         ,\ifx\@currext\@clsextension\else\@classoptionslist,\fi
         \@curroptions,}%
      \ifin@
        \@use@ption
        \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
      \fi
    \fi}%
  \@process@pti@ns}
\@onlypreamble\@process@ptions
```

```
\def\@xprocess@ptions{%
  \ifx\@currext\@clsextension\else
    \@for\CurrentOption:=\@classoptionslist\do{%
      \ifx\CurrentOption\@empty\else
        \@expandtwoargs\in@{,\CurrentOption,}{,\@declaredoptions,}%
        \ifin@
          \@use@ption
          \expandafter\let\csname ds@\CurrentOption\endcsname\@empty
        \fi
      \fi}%
  \fi
  \@process@pti@ns}
\@onlypreamble\@xprocess@ptions
```

The common part of `\ProcessOptions` and `\ProcessOptions*`.

```
\def\@process@pti@ns{%
  \@for\CurrentOption:=\@curroptions\do{%
    \@ifundefined{ds@\CurrentOption}%
      {\@use@ption
       \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`.

```
      \@use@ption}%
```

Clear all the definitions for option code. First set all the declared options to `\relax`, then reset the 'default' and 'empty' options. and the lst of declared options.

```
  \@for\CurrentOption:=\@declaredoptions\do{%
    \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%
```

```
  \let\CurrentOption\@empty
  \let\@fileswith@pti@ns\@@fileswith@pti@ns
  \AtEndOfPackage{\let\@unprocessedoptions\relax}}
\@onlypreamble\@process@pti@ns
```

`\@options` `\@options` is a synonym for `\ProcessOptions*` for upward compatibility with LaTeX2.09 style files.

```
\def\@options{\ProcessOptions*}
\@onlypreamble\@options
```

`\@use@ption` Execute the code for the current option.

```
\def\@use@ption{%
  \@expandtwoargs\@removeelement\CurrentOption
  \@unusedoptionlist\@unusedoptionlist
  \csname ds@\CurrentOption\endcsname}
\@onlypreamble\@use@ption
```

`\ExecuteOptions` `\ExecuteOptions{`⟨*option-list*⟩`}` executes the code declared for each option.

```
\def\ExecuteOptions#1{%
  \def\reserved@a##1\@nil{%
    \@for\CurrentOption:=#1\do{\csname ds@\CurrentOption\endcsname}%
    \edef\CurrentOption{##1}}%
  \expandafter\reserved@a\CurrentOption\@nil}
\@onlypreamble\ExecuteOptions
```

The top-level commands, which just set some parameters then call the internal command, `\@fileswithoptions`.

`\documentclass` The main new-style class declaration.

```
\def\documentclass{%
  \let\documentclass\@twoclasseserror
  \if@compatibility\else\let\usepackage\RequirePackage\fi
  \@fileswithoptions\@clsextension}
\@onlypreamble\documentclass
```

`\documentstyle` 2.09 style class 'style' declaration.

```
\def\documentstyle{%
  \makeatletter\input{latex209.def}\makeatother
  \documentclass}
\@onlypreamble\documentstyle
```

`\RequirePackage` Load package if not already loaded.

```
\def\RequirePackage{%
  \@fileswithoptions\@pkgextension}
\@onlypreamble\RequirePackage
```

`\LoadClass` Load class.

```
\def\LoadClass{%
  \ifx\@currext\@pkgextension
     \@latex@error
      {\noexpand\LoadClass in package file}%
      {You may only use \noexpand\LoadClass in a class file.}%
  \fi
  \@fileswithoptions\@clsextension}
\@onlypreamble\LoadClass
```

`\@loadwithoptions` Pass the current option list on to a class or package. `#1` is `\@`*cls-or-pkg*`extension`, `#2` is `\RequirePackage` or `\LoadClass`, `#3` is the class or package to be loaded.

```
\def\@loadwithoptions#1#2#3{%
  \expandafter\let\csname opt@#3.#1\expandafter\endcsname
     \csname opt@\@currname.\@currext\endcsname
   #2{#3}}
\@onlypreamble\@loadwithoptions
```

`\LoadClassWithOptions` Load class '`#1`' with the current option list.

```
\def\LoadClassWithOptions{%
  \@loadwithoptions\@clsextension\LoadClass}
\@onlypreamble\LoadClassWithOptions
```

`\RequirePackageWithOptions` Load package '`#1`' with the current option list.

```
\def\RequirePackageWithOptions{%
  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
  \@loadwithoptions\@pkgextension\RequirePackage}
\@onlypreamble\RequirePackageWithOptions
```

`\usepackage` To begin with, `\usepackage` produces an error. This is reset by `\documentclass`.

```
\def\usepackage#1#{%
  \@latex@error
    {\noexpand \usepackage before \string\documentclass}%
    {\noexpand \usepackage may only appear in the document
      preamble, i.e.,\MessageBreak
      between \noexpand\documentclass and
      \string\begin{document}.}%
  \@gobble}
\@onlypreamble\usepackage
```

`\NeedsTeXFormat` Check that the document is running on the correct system.

```
\def\NeedsTeXFormat#1{%
  \def\reserved@a{#1}%
  \ifx\reserved@a\fmtname
    \expandafter\@needsformat
  \else
     \@latex@error{This file needs format `\reserved@a'%
       \MessageBreak but this is `\fmtname'}{%
       The current input file will not be processed
       further,\MessageBreak
       because it was written for some other flavor of
       TeX.\MessageBreak\@ehd}%
```

If the file is not meant to be processed by LaTeX 2_ε we stop inputting it, but we do not end the run. We just end inputting the current file.

```
     \endinput \fi}
\@onlypreamble\NeedsTeXFormat
```

```
\def\@needsformat{%
  \@ifnextchar[%]
    \@needsf@rmat
    {}}
\@onlypreamble\@needsformat
```

```
\def\@needsf@rmat[#1]{%
    \@ifl@t@r\fmtversion{#1}{}%
    {\@latex@warning@no@line
        {You have requested release ‘#1’ of LaTeX,\MessageBreak
         but only release ‘\fmtversion’ is available}}}
\@onlypreamble\@needsf@rmat
```

`\zap@space`

`\zap@space foo`⟨*space*⟩`\@empty` removes all spaces from `foo` that are not protected by `{ }` groups.

```
\def\zap@space#1 #2{%
  #1%
  \ifx#2\@empty\else\expandafter\zap@space\fi
  #2}
```

`\@fileswithoptions`

The common part of `\documentclass` and `\usepackage`.

```
\def\@fileswithoptions#1{%
  \@ifnextchar[%]
    {\@fileswith@ptions#1}%
    {\@fileswith@ptions#1[]}}
\@onlypreamble\@fileswithoptions
```

```
\def\@fileswith@ptions#1[#2]#3{%
  \@ifnextchar[%]
  {\@fileswith@pti@ns#1[{#2}]#3}%
  {\@fileswith@pti@ns#1[{#2}]#3[]}}
\@onlypreamble\@fileswith@ptions
```

Then we do some work.

First of all, we define the global variables. Then we look to see if the file has already been loaded. If it has, we check that it was first loaded with at least the current options. If it has not, we add the current options to the package options, set the default version to be `0000/00/00`, and load the file if we can find it. Then we check the version number.

Finally, we restore the old file name, reset the default option, and we set the catcode of `@`.

For classes, we can immediately process the file. For other types, `#2` could be a comma separated list, so loop through, processing each one separately.

```
\def\@fileswith@pti@ns#1[#2]#3[#4]{%
  \ifx#1\@clsextension
    \ifx\@classoptionslist\relax
      \xdef\@classoptionslist{\zap@space#2 \@empty}%
      \def\reserved@a{%
        \@onefilewithoptions#3[{#2}][{#4}]#1%
        \@documentclasshook}%
    \else
      \def\reserved@a{%
        \@onefilewithoptions#3[{#2}][{#4}]#1}%
    \fi
  \else
```

build up a list of calls to `\@onefilewithoptions` (one for each package) without thrashing the parameter stack.

```
    \def\reserved@b##1,{%
      \ifx\@nil##1\relax\else
        \ifx\relax##1\relax\else
         \noexpand\@onefilewithoptions##1[{#2}][{#4}]%
         \noexpand\@pkgextension
        \fi
        \expandafter\reserved@b
      \fi}%
      \edef\reserved@a{\zap@space#3 \@empty}%
      \edef\reserved@a{\expandafter\reserved@b\reserved@a,\@nil,}%
  \fi
  \reserved@a}
\@onlypreamble\@fileswith@pti@ns
```

Have the main argument as #1, so we only need one `\expandafter` above.

```
\def\@onefilewithoptions#1[#2][#3]#4{%
  \@pushfilename
  \xdef\@currname{#1}%
  \global\let\@currext#4%
  \expandafter\let\csname\@currname.\@currext-h@@k\endcsname\@empty
  \let\CurrentOption\@empty
  \@reset@ptions
  \makeatletter
```

Grab everything in a macro, so the parameter stack is popped before any processing begins.

```
  \def\reserved@a{%
    \@ifl@aded\@currext{#1}%
      {\@if@ptions\@currext{#1}{#2}{}%
        {\@latex@error
            {Option clash for \@cls@pkg\space #1}%
            {The package #1 has already been loaded
             with options:\MessageBreak
             \space\space[\@ptionlist{#1.\@currext}]\MessageBreak
             There has now been an attempt to load it
              with options\MessageBreak
             \space\space[#2]\MessageBreak
             Adding the global options:\MessageBreak
             \space\space
                  \@ptionlist{#1.\@currext},#2\MessageBreak
             to your \noexpand\documentclass declaration may fix this.%
             \MessageBreak
             Try typing \space <return> \space to proceed.}}}%
      {\@pass@ptions\@currext{#2}{#1}%

       \global\expandafter
       \let\csname ver@\@currname.\@currext\endcsname\@empty
       \InputIfFileExists
         {\@currname.\@currext}%
         {}%
         {\@missingfileerror\@currname\@currext}%
```

`\@unprocessedoptions` will generate an error for each specified option in a package unless a `\ProcessOptions` has appeared in the package file.

```
    \let\@unprocessedoptions\@@unprocessedoptions
    \csname\@currname.\@currext-h@@k\endcsname
    \expandafter\let\csname\@currname.\@currext-h@@k\endcsname
              \@undefined
    \@unprocessedoptions}

    \@ifl@ter\@currext{#1}{#3}{}%
      {\@latex@warning@no@line
         {You have requested,\on@line,
          version\MessageBreak
            `#3' of \@cls@pkg\space #1,\MessageBreak
          but only version\MessageBreak
           `\csname ver@#1.\@currext\endcsname'\MessageBreak
          is available}}%
    \ifx\@currext\@clsextension\let\LoadClass\@twoloadclasserror\fi
    \@popfilename
    \@reset@ptions}%
  \reserved@a}
\@onlypreamble\@onefilewithoptions
```

`\@@fileswith@pti@ns` Save the definition (for error checking).

```
\let\@@fileswith@pti@ns\@fileswith@pti@ns
\@onlypreamble\@@fileswith@pti@ns
```

`\@reset@ptions` Reset the default option, and clear lists of declared options.

```
\def\@reset@ptions{%
  \global\ifx\@currext\@clsextension
    \let\default@ds\OptionNotUsed
   \else
    \let\default@ds\@unknownoptionerror
  \fi
  \global\let\ds@\@empty
  \global\let\@declaredoptions\@empty}
\@onlypreamble\@reset@ptions
```

70.1 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`
`\@enddocumenthook`

Stuff to appear at the begining or end of the document.

```
\ifx\@begindocumenthook\@undefined
  \let\@begindocumenthook\@empty
\fi
\let\@enddocumenthook\@empty
```

`\g@addto@macro` Globally add to the end of a macro.

```
\long\def\g@addto@macro#1#2{%
  \begingroup
    \toks@\expandafter{#1#2}%
    \xdef#1{\the\toks@}%
  \endgroup}
```

`\AtEndOfPackage`
`\AtEndOfClass`
`\AtBeginDocument`
`\AtEndDocument`

The access functions.

```
\def\AtEndOfPackage{%
  \expandafter\g@addto@macro\csname\@currname.\@currext-h@@k\endcsname}
\let\AtEndOfClass\AtEndOfPackage
\@onlypreamble\AtEndOfPackage
\@onlypreamble\AtEndOfClass
```

```
\def\AtBeginDocument{\g@addto@macro\@begindocumenthook}
\def\AtEndDocument{\g@addto@macro\@enddocumenthook}
\@onlypreamble\AtBeginDocument
```

`\@cls@pkg` The current file type.

```
\def\@cls@pkg{%
  \ifx\@currext\@clsextension
    document class%
  \else
    package%
  \fi}
\@onlypreamble\@cls@pkg
```

`\@unknownoptionerror` Bad option.

```
\def\@unknownoptionerror{%
  \@latex@error
    {Unknown option `\CurrentOption' for \@cls@pkg\space`\@currname'}%
    {The option `\CurrentOption' was not declared in
     \@cls@pkg\space`\@currname', perhaps you\MessageBreak
      misspelled its name.
     Try typing \space <return>
     \space to proceed.}}
\@onlypreamble\@unknownoptionerror
```

`\@@unprocessedoptions` Declare an error for each option, unless a `\ProcessOptions` occured.

```
\def\@@unprocessedoptions{%
  \ifx\@currext\@pkgextension
    \edef\@curroptions{\@ptionlist{\@currname.\@currext}}%
    \@for\CurrentOption:=\@curroptions\do{%
        \ifx\CurrentOption\@empty\else\@unknownoptionerror\fi}%
  \fi}
\@onlypreamble\@unprocessedoptions
\@onlypreamble\@@unprocessedoptions
```

`\@badrequireerror` `\RequirePackage` or `\LoadClass` occurs in the options section.

```
\def\@badrequireerror#1[#2]#3[#4]{%
  \@latex@error
    {\noexpand\RequirePackage or \noexpand\LoadClass
         in Options Section}%
    {The \@cls@pkg\space `\@currname' is defective.\MessageBreak
     It attempts to load `#3' in the options section, i.e.,\MessageBreak
     between \noexpand\DeclareOption and \string\ProcessOptions.}}
\@onlypreamble\@badrequireerror
```

`\@twoloadclasserror` Two `\LoadClass` in a class.

```
\def\@twoloadclasserror{%
  \@latex@error
    {Two \noexpand\LoadClass commands}%
    {You may only use one \noexpand\LoadClass in a class file}}
\@onlypreamble\@twoloadclasserror
```

`\@twoclasseserror` Two `\documentclass` or `\documentstyle`.

```
\def\@twoclasseserror#1#{%
  \@latex@error
    {Two \noexpand\documentclass or \noexpand\documentstyle commands}%
    {The document may only declare one class.}\@gobble}
\@onlypreamble\@twoclasseserror
```

70.2 Providing shipment

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

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

`\filecontents`
`\endfilecontents` This environment implements inline files. The star-form does not write extra comments into the file.

```
\begingroup%
\catcode`\*=11 %
\catcode`\^^M\active%
\catcode`\^^L\active\let^^L\relax%
\catcode`\^^I\active%

\gdef\filecontents{\@tempswatrue\filec@ntents}%
\gdef\filecontents*{\@tempswafalse\filec@ntents}%
```

```
\gdef\filec@ntents#1{%
  \openin\@inputcheck#1 %
  \ifeof\@inputcheck%
    \@latex@warning@no@line%
        {Writing file ‘\@currdir#1’}%

    \chardef\reserved@c15 %
    \ch@ck7\reserved@c\write%
    \immediate\openout\reserved@c#1\relax%
  \else%

    \closein\@inputcheck%
    \@latex@warning@no@line%
            {File ‘#1’ already exists on the system.\MessageBreak%
             Not generating it from this source}%
    \let\write\@gobbletwo%
    \let\closeout\@gobble%
  \fi%
  \if@tempswa%

    \immediate\write\reserved@c{%
      \@percentchar\@percentchar\space%
          \expandafter\@gobble\string\LaTeX2e file ‘#1’^^J%
      \@percentchar\@percentchar\space  generated by the %
        ‘\@currenvir’ \expandafter\@gobblefour\string\newenvironment^^J%
      \@percentchar\@percentchar\space from source ‘\jobname’ on %
         \number\year/\two@digits\month/\two@digits\day.^^J%
      \@percentchar\@percentchar}%
  \fi%
  \let\do\@makeother\dospecials%

  \edef\E{\@backslashchar end\string{\@currenvir\string}}%
  \edef\reserved@b{%
    \def\noexpand\reserved@b%
         ####1\E####2\E####3\relax}%
  \reserved@b{%
    \ifx\relax##3\relax%
```

There was no `\end{filecontents}`

```
      \immediate\write\reserved@c{##1}%
    \else%
```

There was a `\end{filecontents}`, so stop this time.

```
      \edef^^M{\noexpand\end{\@currenvir}}%
      \ifx\relax##1\relax%
      \else%
```

Text before the `\end`, write it with a warning.

```
          \@latex@warning{Writing text ‘##1’ before %
             \string\end{\@currenvir}\MessageBreak as last line of #1}%
        \immediate\write\reserved@c{##1}%
      \fi%
      \ifx\relax##2\relax%
      \else%
```

Text after the `\end`, ignore it with a warning.

```
         \@latex@warning{%
           Ignoring text ‘##2’ after \string\end{\@currenvir}}%
      \fi%
    \fi%
    ^^M}%

  \catcode‘\^^L\active%
  \let\L\@undefined%
  \def^^L{\@ifundefined L^^J^^J^^J}%
  \catcode`\^^I\active%
  \let\I\@undefined%
  \def^^I{\@ifundefined I\space\space}%
  \catcode`\^^M\active%
  \edef^^M##1^^M{%
    \noexpand\reserved@b##1\E\E\relax}}%
\endgroup%
```

```
\begingroup
\catcode`|=\catcode`\%
\catcode`\%=12
\catcode`\*=11
\gdef\@percentchar{%}
\gdef\endfilecontents{|
  \immediate\closeout\reserved@c
  \def\T##1##2##3{|
  \ifx##1\@undefined\else
    \@latex@warning@no@line{##2 has been converted to Blank ##3e}|
  \fi}|
  \T\L{Form Feed}{Lin}|
  \T\I{Tab}{Spac}|
  \immediate\write\@unused{}}
\global\let\endfilecontents*\endfilecontents
\@onlypreamble\filecontents
\@onlypreamble\endfilecontents
\@onlypreamble\filecontents*
\@onlypreamble\endfilecontents*
\endgroup
\@onlypreamble\filec@ntents
```

```
⟨/2ekernel⟩
```

71 After Preamble

Finally we declare a package that allows all the commands declared above to be `\@onlypreamble` to be used after `\begin{document}`.

```
⟨*afterpreamble⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{pkgindoc}
         [1994/10/20 v1.1 Package Interface in Document (DPC)]
\def\reserved@a#1\do\@classoptionslist#2\do\filec@ntents#3\relax{%
  \gdef\@preamblecmds{#1#3}}
\expandafter\reserved@a\@preamblecmds\relax
⟨/afterpreamble⟩
```

File M

lthyphen.dtx

This file contains the code for loading hyphenation patterns into LaTeX. Most of this will end up in a file called `hyphen.ltx`. If you wish to customize your LaTeX 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 LaTeX 2ε.

```
⟨*driver⟩
\documentclass{ltxdoc}
\begin{document}
\DocInput{lthyphen.dtx}
\end{document}
⟨/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 TeX'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.

```
⟨*default⟩
\InputIfFileExists{hyphen.tex}%
  {\message{Loading hyphenation patterns for US english.}%
   \language=0
   \lefthyphenmin=2 \righthyphenmin=3 }%
```

Otherwise, since we cannot do anything without any hyphenation patterns, an error message is printed and the IniTeX run is terminated by invoking `\@@end` (which is the LaTeX 2ε name for TeX's `\end` primitive).

```
  {\errhelp{The configuration for hyphenation is incorrectly
            installed.^^J%
            If you don't understand this error message you need
            to seek^^Jexpert advice.}%
   \errmessage{OOPS! I can't find any hyphenation patterns for
               US english.^^J \space Think of getting some or the
               latex2e setup will never succeed}\@@end}
⟨/default⟩
```

The following example describes the possible contents of a file `hyphen.cfg` that will load both US English and German hyphenation patterns, making the former the default. It sets `\language` to 0 for the US patterns and to 1 for the German patterns. Then `\language` is set to 0 to make this the default and the default values of `\lefthyphenmin` and `\righthyphenmin` are set.

```
\language=0
\input hyphen % (or \input ushyphen1 if the file has been renamed)
\language=1
\input ghyph31
\language=0
\lefthyphenmin=2
\righthyphenmin=3
\endinput
```

Another possibility is to use the package `babel`, by Johannes Braams. That package is distributed with a suitable `hyphen.cfg` file.

File N

ltfinal.dtx

72 Final settings

This section contains the final settings for LaTeX. It initialises some debugging and typesetting parameters, sets the default `\catcode`s and uc/lc codes, and inputs the hyphenation file.

72.1 Debugging

By default, LaTeX shows statistics:

```
⟨*2ekernel⟩
\tracingstats1
```

72.2 Typesetting parameters

`\@lowpenalty`
`\@medpenalty`
`\@highpenalty`

These are penalties used internally.

```
\newcount\@lowpenalty
\newcount\@medpenalty
\newcount\@highpenalty
```

The default values of the picture and `\fbox` parameters:

```
\unitlength = 1pt
\fboxsep = 3pt
\fboxrule = .4pt
```

The saved value of TeX's `\maxdepth`:

```
\@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}`

```
\vsize = 1000pt
\@colroom = \vsize
\@colht = \vsize
```

Initialise `\textheight` `\textwidth` and page style, to avoid internal errors if they are not set by the class.

```
\textheight=.5\maxdimen
\textwidth=\textheight
\ps@empty
```

72.3 Lccodes for hyphenation

We set things up so that hypehnation 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.

```
\def\reserved@a#1#2{%
  \@tempcnta#1\relax
  \@tempcntb#2\relax
  \reserved@b
}
\def\reserved@b{%
  \ifnum\@tempcnta>\@tempcntb\else
    \reserved@c\@tempcnta
    \advance\@tempcnta\@ne
    \expandafter\reserved@b
```

```
   \fi
}
```

Depending on the TEX version, we might not be allowed to do this for non-ASCII characters.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by -"20
   \uccode#1=\count@
   \lccode#1=#1
}
\reserved@a{‘\a}{‘\z}
\ifnum\inputlineno=\m@ne\else
  \reserved@a{"A0}{"BC}
  \reserved@a{"E0}{"FF}
\fi
```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by "20
   \uccode#1=#1
   \lccode#1=\count@
   \sfcode#1=999
}
\reserved@a{‘\A}{‘\Z}
\ifnum\inputlineno=\m@ne\else
  \reserved@a{"80}{"9C}
  \reserved@a{"C0}{"DF}
\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.

```
\uccode‘\^^Y=‘\I     % dotless i
\lccode‘\^^Y=‘\^^Y   % dotless i
\uccode‘\^^Z=‘\J     % dotless j, ae in OT1
\lccode‘\^^Z=‘\^^Z   % dotless j, ae in OT1
\ifnum\inputlineno=\m@ne\else
  \lccode‘\^^9d=‘\i    % dotted I
  \uccode‘\^^9d=‘\^^9d % dotted I
  \lccode‘\^^9e=‘\^^9e % d-bar
  \uccode‘\^^9e=‘\^^d0 % d-bar
\fi
```

Finally here is one that helps hyphenation in the OT1 encoding.

```
\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).

```
\lccode‘\- =‘\-   % default hyphen char
\lccode 127=127   % alternate hyphen char
\lccode 23 =23    % textcompwordmark in T1
```

72.4 Hyphenation

The following code will be compiled into the format file. It checks for the existance of `hyphen.cfg` in inputs that file if found. Otherwise it inputs `hyphen.ltx`. Note that these are loaded in *before* the `\catcode`s are set, so local hyphenation files can use 8-bit input.

We try to load the customized hyphenation description file.

```
\InputIfFileExists{hyphen.cfg}
           {\typeout{===========================================^^J%
                      Local configuration file hyphen.cfg used^^J%
```

```
                     ============================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
           }
           {\input{hyphen.ltx}}
\let\@addtofilelist\@gobble
```

72.5 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.

```
% \changes{v1.1c}{2000/08/23}{Fix typo in warning}
\ifdim \font@submax >\z@
   \@font@warning{Size substitutions with differences\MessageBreak
                 up to \font@submax\space have occurred.\MessageBreak
                \MessageBreak
                Please check the transcript file
                carefully\MessageBreak
                and redo the format generation if necessary!
                \@gobbletwo}%
   \errhelp{Only stopped, to give you time to
            read the above message.}
   \errmessage{}
```

We reset the macro. Otherwise every user will get a warning on every job.

```
\def\font@submax{0pt}
\fi
```

72.6 Input encoding

We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
\def\reserved@a#1#2{%
   \@tempcnta#1\relax
   \@tempcntb#2\relax
   \reserved@b
}
\def\reserved@b{%
   \ifnum\@tempcnta>\@tempcntb\else
      \reserved@c\@tempcnta
      \advance\@tempcnta\@ne
      \expandafter\reserved@b
   \fi
}
```

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.

```
\catcode`\ =10
\catcode`\#=6
\catcode`\$=3
\catcode`\%=14
\catcode`\&=4
\catcode`\\=0
\catcode`\^=7
\catcode`\_=8
\catcode`\{=1
\catcode`\}=2
\catcode`\~=13
\catcode`\@=11
\catcode`\^^I=10
\catcode`\^^J=12
\catcode`\^^L=13
```

```
\catcode`\^^M=5
```

Set the ‘other’ catcodes.

```
\def\reserved@c#1{\catcode#1=12\relax}
\reserved@c{`\!}
\reserved@c{`\"}
\reserved@a{`\'}{`\?}
\reserved@c{`\[}
\reserved@c{`\]}
\reserved@c{`\`}
\reserved@c{`\|}
```

Set the ‘letter’ catcodes.

```
\def\reserved@c#1{\catcode#1=11\relax}
\reserved@a{`\A}{`\Z}
\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.

```
\def\reserved@c#1{\catcode#1=15\relax}
\reserved@a{0}{`\^^H}
\reserved@c{`\^^K}
\reserved@a{`\^^N}{31}
%\ifnum\inputlineno=\m@ne
  \catcode"7F=15
%\else
%  \reserved@a{"7F}{"FF}
%\fi
```

72.7 Lccodes and uccodes

We now again set up the default (T1) uc/lccodes. The lower case characters need their `\uccode` and `\lccode` values set. Some of this is a repeat of the set-up before loading hyphenation files. Depending on the TEX version, we might not be allowed to do this for non-ASCII characters.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by -"20
   \uccode#1=\count@
   \lccode#1=#1
}
\reserved@a{`\a}{`\z}
\ifnum\inputlineno=\m@ne\else
   \reserved@a{"A0}{"BC}
   \reserved@a{"E0}{"FF}
\fi
```

The upper case characters need their `\uccode` and `\lccode` values set, and their `\sfcode` set to 999.

```
\def\reserved@c#1{%
   \count@=#1\advance\count@ by "20
   \uccode#1=#1
   \lccode#1=\count@
   \sfcode#1=999
}
\reserved@a{`\A}{`\Z}
\ifnum\inputlineno=\m@ne\else
   \reserved@a{"80}{"9C}
   \reserved@a{"C0}{"DF}
\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.

```
\uccode`\^^Y=`\I     % dotless i
\lccode`\^^Y=`\^^Y   % dotless i
\uccode`\^^Z=`\J     % dotless j, ae in OT1
\lccode`\^^Z=`\^^Z   % dotless j, ae in OT1
\ifnum\inputlineno=\m@ne\else
  \lccode`\^^9d=`\i    % dotted I
  \uccode`\^^9d=`\^^9d % dotted I
  \lccode`\^^9e=`\^^9e % d-bar
  \uccode`\^^9e=`\^^d0 % d-bar
\fi
```

Finally here is one that helps hyphenation in the OT1 encoding.

```
\lccode`\^^[=`\^^[   % oe in OT1
```

`\MakeUppercase`
`\MakeUppercase`
`\@uclclist`

And whilst we're doing things with uc/lc tables, here are two commands to upper- and lower-case a string.

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.

```
\DeclareRobustCommand{\MakeUppercase}[1]{{%
      \def\i{I}\def\j{J}%
      \def\reserved@a##1##2{\let##1##2\reserved@a}%
      \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
      \protected@edef\reserved@a{\uppercase{#1}}%
      \reserved@a
   }}
\DeclareRobustCommand{\MakeLowercase}[1]{{%
      \def\reserved@a##1##2{\let##2##1\reserved@a}%
      \expandafter\reserved@a\@uclclist\reserved@b{\reserved@b\@gobble}%
      \protected@edef\reserved@a{\lowercase{#1}}%
      \reserved@a
   }}
\def\@uclclist{\oe\OE\o\O\ae\AE
      \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:

```
\protected@edef\MakeUppercase#1{\MakeUppercase{#1}}
\protected@edef\MakeLowercase#1{\MakeLowercase{#1}}
```

72.8 Applying Patch files

Between major releases, small patches will be distributed in files `ltpatch.ltx` which must be added at this point.

```
\IfFileExists{ltpatch.ltx}
  {\typeout{==================================^^J%
```

```
           Applying patch file ltpatch.ltx^^J%
          ================================}
 \def\fmtversion@topatch{unknown}
 \input{ltpatch.ltx}
 \ifx\fmtversion\fmtversion@topatch
   \ifx\patch@level\@undefined
      \typeout{^^J^^J^^J%
       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
       !! Patch file `ltpatch.ltx' not suitable for this^^J%
       !! version of LaTeX.^^J^^J%
       !! Please check if initex found an old patch file:^^J%
       !! --- if so, rename it or delete it, and redo the^^J%
       !! initex run.^^J%
       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
      \batchmode \@@end
    \else
```

The code below adds the 'patch level' string to the first `\typeout` in the startup banner.

```
      \def\fmtversion@topatch{0}%
      \ifx\fmtversion@topatch\patch@level\else
        \def\reserved@a\typeout##1##2\reserved@a{%
               \typeout{##1 patch level \patch@level}##2}
        \everyjob\expandafter\expandafter\expandafter{%
         \expandafter\reserved@a\the\everyjob\reserved@a}
        \let\reserved@a\relax
        \the\everyjob
      \fi
    \fi
 \else
    \typeout{^^J^^J^^J%
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J%
   !! Patch file `ltpatch.ltx' (for version <\fmtversion@topatch>)^^J%
   !! is not suitable for version <\fmtversion> of LaTeX.^^J^^J%
   !! Please check if initex found an old patch file:^^J%
   !! --- if so, rename it or delete it, and redo the^^J%
   !!     initex run.^^J%
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!^^J}%
     \batchmode \@@end
 \fi
 \let\fmtversion@topatch\relax
}{}
```

72.9 Freeing Memory

`\reserved@a`
`\reserved@b`

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.*

```
\let\reserved@a\@filelist
\let\reserved@b=\@undefined
\let\reserved@c=\@undefined
\let\reserved@d=\@undefined
\let\reserved@e=\@undefined
\let\reserved@f=\@undefined
```

`\toks`

```
\toks0{}
\toks2{}
\toks4{}
\toks6{}
\toks8{}
```

`\errhelp` Empty the error help message, which may have some rubbish:

```
\errhelp{}
```

72.10 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`.

```
\def\@providesfile#1[#2]{%
    \wlog{File: #1 #2}%
    \expandafter\xdef\csname ver@#1\endcsname{#2}%
  \endgroup}
```

`\@filelist`
`\@addtofilelist` 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.

```
\let\@filelist\@gobble
\def\@addtofilelist#1{\xdef\@filelist{\@filelist,#1}}%
```

72.11 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.

```
\makeatother
\errorstopmode
\dump
⟨/2ekernel⟩
```

File O
ltpatch

Things we did wrong…

```
%%%%
%%%% Patch file for the LaTeX2e kernel dated 2011/06/27
%%%% (2011/06/27)

\def\fmtversion@topatch{2011/06/27} % This patch will not work with
                                    % any other release.

\def\patch@level{0}




%%%%%%%%%%%%%%%
\iffalse

\typeout{%
^^J%
*********************************************^^J%
ltpatch.ltx has fixed certain problems with the ‘kernel’ of LaTeX.^^J%
Certain other files in the LaTeX distribution have also been updated^^J%
since the last release (list correct as of 2011/06/27):^^J%
base/xxxxxxx.dtx..............(patch 1)^^J%
unpacked/yyyyyyy.cls..........(patch 1)^^J%
^^J%
See the file patches.txt for more details.^^J%
*********************************************}

\fi

\endinput
```

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. 215

1989/04/10 ltfssbas.dtx v1.0a
- General: Starting with version numbers! `\ifmmode` added in `\math@group` 107

1989/04/10 ltfssbas.dtx v1.0b
- General: `\preload@sizes` added. 107
- `\wrong@fontshape` changed to define substitution font/shape macro. 107

1989/04/10 ltfssini.dtx v1.0a
- General: Starting with version numbers `\newif` for `\@tempswa` added since this switch is unkown at the time when this file is read in. (latex.tex is loaded later.) `\math@famname` changed to `\math@version`. 168

1989/04/14 ltfssbas.dtx v1.0c
- General: More documentation added. 107

1989/04/15 ltfssini.dtx v1.0b
- General: `\mathfontset` renamed to `\mathversion`. 168

1989/04/19 ltfssbas.dtx v1.0d
- General: Even more doc. 107

1989/04/21 ltfssbas.dtx v1.0e
- General: Documentation is fun! Parameters of `\define@mathalphabet` changed. 107

1989/04/21 ltfssini.dtx v1.0c
- General: Changed to conform to fam.tex. 168

1989/04/23 ltfssbas.dtx v1.0f
- General: % in `\getanddefinefonts` added. 107

1989/04/26 ltfssini.dtx v1.0d
- General: `\xpt` added. 168

1989/04/27 ltfssbas.dtx v1.0g
- General: Documentation revised. 107

1989/04/27 ltfssini.dtx v1.0e
- General: Definitions of LaTeX symbols corrected. 168

1989/04/29 ltfssbas.dtx v1.0h
- General: Documented problem with `\halign`, and `\noalign` 107
- `\mathversion`: Test if version defined added. 114

1989/04/29 ltfssbas.dtx v1.0i
- General: Removed the `\halign` `\noalign` correction (wasn't bugfree) 107

1989/04/29 ltfssini.dtx v1.0f
- General: Corrections to LaTeX tabular env. added. 168

1989/05/01 ltfssbas.dtx v1.0j
- General: Default for `\baselinestretch` added. 107

1989/05/22 ltfssbas.dtx v1.0k
- General: Lines longer than 72 characters folded. 107

1989/05/22 ltfssini.dtx v1.0g
- General: Lines shortened to 72 characters 168

1989/09/14 ltfssbas.dtx v1.0m
- General: Global replacement: `\group` to `\mathgroup` 107
- `\mathversion`: Corrected typo: `\endscname` to `\endcsname`. 114

1989/11/07 ltfssini.dtx v1.0i
- General: All family, series, and shape names abbreviated. 168

1989/11/08 ltfssbas.dtx v1.0o
- General: First parameter of `\define@mathalphabet` and `\define@mathgroup` changed from string to control sequence. 107

1989/11/14 ltfssbas.dtx v1.0p
- `\math@version`: Math version prefix 'mv@' added. 114

1989/11/19 ltfssbas.dtx v1.0q
- `\define@newfont`: Group added. 116
- `\wrong@fontshape`: Instead of calling `\family\default@family`, etc. we directly set `\f@family`, etc. 119

1989/11/22 ltfssbas.dtx v1.0r
- `\math@version`: `\def` → `\edef` for `\math@version`. 114

1989/11/25 ltfssbas.dtx v1.0s
- General: All `\edef\font@name` changed to `\xdef\font@name`. Necessary after introduction of `\begingroup/\endgroup` in v1.0q. 107
- extra// → + in `\extra@def`. 107

1989/11/26 ltfssbas.dtx v1.0t
- `\select@group`: `\bgroup/\egroup` changed to `\begingroup/\endgroup` to avoid empty Ord atom on math list. 120

1989/12/02 ltfssini.dtx v1.1b
- General: `\rmmath` renamed to `\mathrm` 168

1989/12/03 ltfssini.dtx v1.1c
- General: Some internal macros renamed to make them inaccessible. 168

1989/12/05 ltfssbas.dtx v1.0u
- `\addto@hook`: `\addto@hook` added. 123

1989/12/05 ltfsstrc.dtx v1.0u fam.dtx
- `\every@math@size`: Hook `\every@size` added. 131

1989/12/13 ltfsstrc.dtx v1.0f
- `\use@mathgroup`: `\expandafter` added before final `\fi`. 134

1989/12/16 ltfssbas.dtx v1.1a
- `\select@group`: `\relax` in front added. 120
  - Now four arguments. 120
  - Redefinition of alphabet now simpler. 121
  - Usage of '=' macro added. 121

1989/12/16 ltfsstrc.dtx v1.1a
- `\selectfont`: Changed order of calls. 128
- `\use@mathgroup`: Redefinition of alphabet now simpler. 133
  - Usage of '=' macro added. 133

1990/01/18 ltfsstrc.dtx v1.0h
- General: `\tracingfonts` meaning changed. 125

1990/01/20 ltfssbas.dtx v1.2a
- `\math@bgroup`: Def. placed in this file. 122
- `\math@egroup`: Def. placed in this file. 122
- `\select@group`: Def for alph id changed. 121

1990/01/21 ltfssbas.dtx v1.2b
- `\select@group`: Code moved to `\use@mathgroup`. 121

1990/01/21 ltfsstrc.dtx v1.2b
- `\use@mathgroup`: Macro added to allow cleaner interface. 133

1990/01/23 ltfssbas.dtx v1.2c
- General: `\no@version@warning` renamed to `\no@alphabet@error`. 107
  - Macro `\no@alphabet@help` added 107
- `\no@alphabet@error`: Changed to error call 107

1990/01/25 ltfssini.dtx v1.1e
- `\nfss@text`: Macro added. 170

1990/01/27 ltfssbas.dtx v1.2d
- `\DeclarePreloadSizes`: Font identifier set to `\relax`. 111

1990/01/28 ltfssbas.dtx v1.2e
- `\mathgroup`: `\newfam` let to `\new@mathgroup`. 108

1990/01/28 ltfssbas.dtx v1.2f
- `\define@newfont`: Added call to `\curr@fontshape` macro to allow substitution. 116
- `\wrong@fontshape`: Warning message slightly changed. 119

1990/01/28 ltfssini.dtx v1.2b
- General: Call to `\@nomath` added. 168

1990/02/08 ltfssini.dtx v1.1g
- General: Protected the commands `\family`, `\series`, `\shape`, `\size`, `\selectfont`, and `\mathversion`. 168

1990/02/16 ltfssbas.dtx v1.2g
- General: Support for changes of `\baselineskip` without changing the size. 107
- `\math@version`: `\@nomath` added. 114

1990/02/16 ltfsstrc.dtx v1.0i
- `\selectfont`: Changed `\f@size` to `\lcl@currsize` (see fam file). 129

1990/02/18 ltfsstrc.dtx v1.0j
- General: Redefine unprotected version `\p@selectfont` instead of `\selectfont`. 128

1990/03/14 ltfsstrc.dtx v1.0k
- General: Added code for TeX3. 125
- `\extract@font`: Added code for TeX3. 128
- `\selectfont`: Added code for TeX3. 129

1990/03/30 ltfssbas.dtx v1.2h
- `\math@egroup`: Changed to have one arg. 122

1990/03/30 ltfsstrc.dtx v1.2h
- `\use@mathgroup`: Third argument removed (see `\math@egroup`). 133

1990/04/01 ltfssbas.dtx v1.2i
- General: Code added from tracefnt.dtx. 107
  - Support for TeX3. 107

1990/04/01 ltfsstrc.dtx v1.0l
- General: Part of code moved to fam.dtx. 125
- `\tracingfonts`: Check if `\tracingfonts` already defined. 126

1990/04/01 ltfsstrc.dtx v1.0o
- `\tracingfonts`: Check if `\tracingfonts` defined removed again. 126

1990/04/02 ltfssini.dtx v1.1i
- General: `\input` of files now handled by docstrip. 168

1990/04/05 ltfsstrc.dtx v1.0m
- `\selectfont`: Call `\tracingon` only if `\tracingfonts` greater than 3. 128

1990/05/05 ltfsstrc.dtx v1.0n
- `\selectfont`: `\tracingon` with new syntax. 128

1990/06/23 ltfssini.dtx v1.1k
- `\nfss@text`: Changed to `\mbox`. 170

1990/06/24 ltfssbas.dtx v1.2j
- `\DeclarePreloadSizes`: Missing percent added. 111

1990/06/24 ltfsstrc.dtx v1.0o
- `\baselinestretch`: Moved to tracefnt.dtx. 131
- `\getanddefine@fonts`: `\Adding` tracing code. 134
  - `\Macro` moved from fam.dtx. 134
  - Adding debug code. 134

- `\use@mathgroup`: Tracing code added. 133

1990/06/30 ltfssbas.dtx v1.2l
- `\showhyphens`: Macro added. . . . 123

1990/06/30 ltfsstrc.dtx v1.0p
- `\use@mathgroup`: Added `\relax` after math group number. . . . 134

1990/07/07 ltfsstrc.dtx v1.0q
- `\getanddefine@fonts`: Group number added to tracing. . . 134
- `\math@egroup`: Tracing code added. 134
- `\use@mathgroup`: Group number added to tracing. 133

1990/08/27 ltfsstrc.dtx 1.0r
- `\type@restoreinfo`: Some extra tracing info. 130

1990/08/27 ltfsstrc.dtx v1.0r
- `\getanddefine@fonts`: Correcting missing name after `\tracingon`. 134

1991/03/28 ltfssini.dtx v1.1m
- `\copyright`: Extra braces added. 170

1991/03/30 ltfssini.dtx v1.2g
- `\newfont`: Definition added. 169
- `\symbol`: Definition added. 169

1991/07/24 ltmiscen.dtx LaTeX2.09
- `\@verbatim`: Added `\penalty\interlinepenalty` to definition of `\par` so that `\samepage` works 207

1991/08/14 ltmath.dtx LaTeX2.09
- `\cases`: (RmS) inserted extra braces around entry for NFSS 212

1991/08/14 ltpictur.dtx LaTeX2.09
- General: (RmS) inserted extra braces around entry for NFSS 263

1991/08/14 ltthm.dtx LaTeX2.09
- `\@endtheorem`: Moved `\itshape` after `\item` to make it work with NFSS 281

1991/08/26 ltfssini.dtx v1.1n
- `\p@reset@font`: Macro introduced 170

1991/08/26 ltmiscen.dtx LaTeX2.09
- `\@verbatim`: `\@@par` added 207

1991/08/26 ltpictur.dtx LaTeX2.09
- `\endpicture`: (RmS & FMi) extra boxing level around `\@picbox` to guard against unboxing in math mode (proposed by John Hobby) 262

1991/08/26 ltplain.dtx LaTeX209
- `\tracingall`: Added `\errorcontextlines=\maxdimen`, suggested by J. Schrod 20

1991/09/29 ltboxes.dtx LaTeX2.09
- `\@mpfootnotetext`: (RmS) added `\reset@font` 239

1991/09/29 ltfloat.dtx LaTeX2.09
- `\@footnotetext`: (RmS) added `\reset@font` 304

1991/09/29 ltmath.dtx LaTeX2.09
- `\@eqnnum`: RmS: `\reset@font` added. 214

1991/09/29 ltsect.dtx LaTeX2.09
- `\@dottedtocline`: (RmS) added `\reset@font` for page number 290

1991/10/17 ltcntrl.dtx LaTeX209
- `\@tfor`: (Rms) `\xdef` replaced by `\def` (See FMi's array.doc) . . 37

1991/10/25 ltbibl.dtx LaTeX2.09
- `\@citex`: added `\reset@font`, suggested by Bernd Raichle. . . . 308

1991/11/01 ltfloat.dtx LaTeX2.09
- `\footnote`: (RmS) Added `\let\protect\noexpand` in `\footnote`, `\footnotemark`, and `\footnotetext`, since `\xdef` is used 303

1991/11/04 ltlists.dtx LaTeX2.09
- `\makelabel`: (RmS) added default definition for `\makelabel`, to produce an error message. . . 229

1991/11/04 ltplain.dtx RmS
- General: Removed `\itemitem` since never needed/useful with LaTeX. 19

1991/11/06 ltbibl.dtx LaTeX2.09
- `\@citex`: added code to remove a leading blank 308

1991/11/13 ltbibl.dtx LaTeX2.09
- `\@bibitem`: Changed counter enumi to enumiv, as it says in the comment above 308

1991/11/21 ltfssini.dtx v1.1o
- `\p@reset@font`: Added extra braces for robustness. 170
- Changed to protected version of macro. 170

1991/11/22 ltfloat.dtx LaTeX2.09
- `\footnote`: (RmS) Added `\let\protect\noexpand` in `\@xfootnote`, `\@xfootnotemark`, and `\@xfootnotetext` 303

1991/11/22 ltlists.dtx LaTeX2.09
- `\@item`: (RmS) Changed second call to `\makelabel` to `\unhbox\@tempboxa`. Avoids problems with side effects in `\makelabel` and is more efficient. . 229

1991/11/27 ltfssbas.dtx v1.3a
- General: All `\family`, `\shape` etc. renamed to `\fontfamily` etc. 107

1991/11/27 ltfssini.dtx v1.2a
- General: All `\family`, `\shape` etc. renamed to `\fontfamily` etc. 168

1992/01/06 ltfssini.dtx v1.2c
- General: added slitex code 168

1992/01/10 ltbibl.dtx LaTeX2.09
- `\@bibitem`: Changed `\c@enumiv` to `\value` of `\@listctr` 308

- 1992/01/10 ltmath.dtx LaTeX2.09
  - `equation`: RmS: put `\hbox` around `\@eqnnum` to typeset the equation number in text mode (as in the eqnarray env.) 214
- 1992/01/10 ltthm.dtx LaTeX2.09
  - `\@othm`: (RmS) Check for existence of theorem environment 281
- 1992/01/14 ltbibl.dtx LaTeX2.09
  - `\@biblabel`: removed `\hfill` . . . 310
- 1992/01/14 ltsect.dtx 0.0
  - `\@starttoc`: (RmS) added `\immediate` to `\openout` as all `\write` commands are also executed `\immediate` 288
- 1992/02/26 ltbibl.dtx LaTeX2.09
  - `\@lbibitem`: Added `\hfill` to restore left-alignment of bibliography labels in alpha style . . 308
- 1992/03/18 ltdefns.dtx LaTeX209
  - General: (RMS) changed input channel from 0 to `\@inputcheck` to avoid conflicts with other channels allocated by `\newread` 24
- 1992/03/18 ltfloat.dtx LaTeX2.09
  - `\@xympar`: (RmS) added `\global\@ignorefalse` 300
  - `\end@float`: (RmS) changed `\@esphack` to `\@Esphack` 296
- 1992/03/18 ltlists.dtx 0.0
  - General: RmS: added `\@nmbrlistfalse` 226
- 1992/03/18 ltmiscen.dtx LaTeX2.09
  - `\begin`: Changed `\@ignoretrue` to `\@ignorefalse` (as documented) 205
- 1992/03/21 ltfssini.dtx v1.2d
  - General: Renamed `\text` to `\nfss@text` to make it internal. 168
- 1992/05/12 ltfssbas.dtx v1.3c
  - `\extract@alph@from@version`: Macro added. 121
  - `\select@group`: Added call to `\extract@alph@from@version`. . 121
- 1992/07/26 ltfssbas.dtx v1.9a
  - `\curr@fontshape`: 116
  - `\DeclareFontShape`: Introduced `\DeclareFontShape` 108
  - `\define@newfont`: 116
  - `\math@fonts`: 120
  - `\select@group`: 120, 121
  - `\split@name`: Added splitting into `\f@encoding`. 116
  - `\wrong@fontshape`: 119
- 1992/07/26 ltfsstrc.dtx v2.0b
  - `\s@fct@`: 142
  - `\s@fct@sub`: 143
  - `\selectfont`: 129
  - `\try@simple@size`: 136, 137
  - `\try@size@range`: 139
  - `\use@mathgroup`: 134
- 1992/08/14 ltbibl.dtx LaTeX2.09
  - `\@citex`: added missing argument braces around `\hbox`, found by Ed Sznyter 308
- 1992/08/14 ltboxes.dtx LaTeX209
  - `\endminipage`: (RmS) replaced `\vskip-\lastskip` by `\unskip` (proposed by FMi) 239
- 1992/08/17 ltbibl.dtx LaTeX2.09
  - `\@citex`: simplified code for removing leading blanks in citation key (proposed by Frank Jensen and Kresten Krab Thorup) . 308
- 1992/08/19 ltsect.dtx 0.0
  - `\@xsect`: (RmS) corrected bug: stretch and shrink in argument to `\hskip` previously not negated 286
- 1992/08/19 ltthm.dtx LaTeX2.09
  - `\@othm`: (RmS) Changed error message to complain about undefined counter 281
- 1992/08/20 ltfssini.dtx v1.4b
  - `\@setsize`: Added `\@currsize`. . 169
- 1992/08/24 ltdefns.dtx LaTeX209
  - `\@ifnextchar`: (Rms) `\@ifnextchar` didn't work if its first argument was an equal sign. 32
- 1992/08/24 ltmiscen.dtx LaTeX2.09
  - `\begin`: Added code to `\begin` to remember line number. Used by `\@badend` to display position of non-matching `\begin`. 205
  - `\verb`: Changed `\verb` and `\@sverb` to work correctly in math mode 208
- 1992/08/25 ltsect.dtx LaTeX2.09
  - `\@sect`: (FMi) replaced explicit setting of `\@svsec` by call to `\@seccntformat` 285
- 1992/09/18 ltlists.dtx LaTeX2.09
  - General: (RmS) Added warning if `\item` is used in math mode 227
- 1992/09/18 lttab.dtx LaTeX2.09
  - `\@array`: Changed `\par` to `\@empty` to avoid starting new row e.g. after `\hline` 251
- 1992/09/19 ltfsstrc.dtx v2.0c
  - `\try@simple@size`: 136
- 1992/09/21 ltfssini.dtx v1.4d
  - `\not@math@alphabet`: Macro defined. 169
- 1992/09/22 ltfssbas.dtx v1.91a
  - General: Introduced `\tf@size` for math size. 107
- 1992/09/22 ltfsstrc.dtx v2.1a
  - `\getanddefine@fonts`: Introduced `\tf@size` for math size. 134
- 1992/11/13 ltfssini.dtx v?
  - `\hexnumber@`: Made expandable. . 170
- 1992/11/23 ltcounts.dtx LaTeX209
  - `\stepcounter`: Replaced `{}` in `\stepcounter` by `\begingroup`

\endgroup to avoid adding an empty ord in math mode . . . 104

1992/11/26 ltboxes.dtx LaTeX2.09
\@mpfootnotetext: (RmS) added protection for \edef 239

1992/11/26 ltfloat.dtx LaTeX2.09
\@footnotetext: (RmS) added protection for \edef 304
\footnote: (RmS) Changed all to ‘def‘protect‘noexpand‘protect‘noexpand 303

1992/12/03 ltfssini.dtx v?
\hexnumber@: Make it accept counters. 170

1993/03/08 preload.dtx v2.0b
General: Added 12pt preloads . . 188

1993/03/18 ltfssbas.dtx v2.0c
General: Changed all \@tempdima in \@tempdimb to avoid killing \numberline 107

1993/03/18 ltfsstrc.dtx v2.1b
General: Changed all \@tempdima in \@tempdimb to avoid killing \numberline 125
Changed all \@tempdimb in \@tempdimx to avoid killing \numberline 125

1993/03/18 ltfsstrc.dtx v2.1c
\DeclareSizeFunction: Added all args to avoid blanks problems 139

1993/04/09 lterror.dtx v1.0e
\@latexerr: Mention The Companion 42

1993/04/11 lterror.dtx v1.0f
\@latexerr: Remove setting of errorcontextlines 42

1993/05/05 ltfntcmd.dtx v2.0b
General: Removed all LaTeX related cmds 191

1993/05/16 ltfssbas.dtx v2.0e
\showhyphens: Use \reset@font 123

1993/07/16 ltfsstrc.dtx v2.1h
General: Changed layout of info messages 125

1993/07/17 ltoutenc.dtx 1.0d
General: changed \catcoding @ . 72

1993/08/03 ltmiscen.dtx LaTeX2.09
\enddocument: Changed redefinition of \global to redefinition of \@setckpt. 203

1993/08/05 ltpictur.dtx LaTeX2.09
\circle: (RMS) Added error message if \circle is used in math mode. 275

1993/08/05 ltsect.dtx LaTeX2.09
\@sect: (RmS) Made sure that \protect works correctly in expansion of \the counter . . . 285

1993/08/05 ltspace.dtx LaTeX2e
\@hspace: (RmS) Removed superfluous \leavevmode in \@hspace and \@hspacer, as suggested by CAR. 57

1993/08/05 lttab.dtx latex2e
\tabular*: Replaced \expandafter\def by \@namedef. 251

1993/08/06 ltbibl.dtx LaTeX2.09
\@citex: Moved writing to .aux file in loop over citation keys so that leading blanks are removed there as well. 308

1993/08/13 ltoutenc.dtx 1.0f
General: Protected against active @ sign. 72

1993/08/13 preload.dtx v2.0c
General: Added \relax at end of font names. 189

1993/08/16 ltoutenc.dtx 1.0g
General: Needs space after \string 72

1993/08/18 ltfssdcl.dtx v2.0e
\new@mathversion: Exchanged names of encodings in warning message of \SetSymbolFont. 154

1993/09/02 ltfsstrc.dtx v2.1i
General: Corrected name of sgen size function. 125

1993/09/03 ltmiscen.dtx LaTeX2.09
\verbatim@nolig@list: Replaced \@noligs by extensible list . 208

1993/09/07 ltmiscen.dtx LaTeX2.09
\verb@balance@group: (RmS) Changed definition of \verb so that it detects a missing second delimiter. 208

1993/09/08 ltmiscen.dtx LaTeX2.09
\enddocument: Added warning in case of undefined references. 203

1993/09/15 ltfssbas.dtx v2.0g
\DeclareFontEncoding: Corrected: \default@T to \default@M. . 110

1993/09/15 ltfsstrc.dtx v2.1j
General: Corrected spelling of \noxpand. 125

1993/09/19 lterror.dtx LaTeX2.09
\@invalidchar: (RmS) Error message for invalid input characters. 45

1993/11/02 ltmath.dtx LaTeX2.09
General: RmS: Corrected description of \@eqnsel, moved \@eqnsel accordingly and removed extra \tabskip assignment. 215

1993/11/03 ltmath.dtx LaTeX2e
General: RmS: Initialized \everycr to empty 215

1993/11/03 ltpictur.dtx LaTeX2.09
General: (RmS) changed \halign to \ialign to initialize \tabskip and \everycr 263

1993/11/11 ltfssini.dtx v2.1a
\normalfont: Macro added 170

1993/11/11 ltfsstrc.dtx v2.2a
- General: Option concept added for LaTeX2e 125

1993/11/14 ltclass.dtx v0.2a
- `\@currext`: Name changed from `\@currextension` 366
- `\@fileswithoptions`: Moved resetting of `\default@ds`, `\ds@` and `\@declaredoptions` here, from the end of `\ProcessOptions`. 373
- `\@reset@ptions`: macro added 374
- `\AtEndDocument`: Included extension in the generated macro name for package and class hooks. 374
- `\documentstyle`: Added `\RequirePackage` `\@unusedoptionlist` stuff. 370
- `\g@addto@macro`: Made global 374
- `\NeedsTeXFormat`: made more robust for alternative syntax for other formats. 371
- `\ProcessOptions*`: Optimise ‘empty option’ code. 369
  - Stop adding the global option list inside class files. 369

1993/11/15 ltclass.dtx v0.2b
- `\documentstyle`: Modified to match `\ProcessOption*` 370
- `\ProcessOptions*`: Star form added. 369

1993/11/17 ltclass.dtx v0.2c
- `\@@fileswith@pti@ns`: Macro added 374
- `\@badrequireerror`: Macro added 375
- `\@fileswithoptions`: Added trap for two `\LoadClass` commands. 374
- `\@twoloadclasserror`: Macro added 375
- `\CurrentOption`: Name changed from `\@curroption` 366
- `\DeclareOption*`: Error checking added 368
- `\NeedsTeXFormat`: Name changed from `\NeedsFormat` 371
- `\ProcessOptions*`: restoring `\@fileswith@pti@ns` added. 369

1993/11/18 ltclass.dtx v0.2d
- `\documentstyle`: Modified `\RequirePackage` stuff. 370
- `\ExecuteOptions`: Use `\CurrentOption` not `\reserved@a` 370
- `\NeedsTeXFormat`: `\fmtname` `\fmtversion` not `\@`... 371

1993/11/21 ltfiles.dtx LaTeX2e
- `\@missingfileerror`: Stop infinite looping on `\@er@ext` 66

1993/11/21 ltmiscen.dtx v0.9a
- `\@verbatim`: use `\verbatim@font` instead of `\tt` 207
- `\verb`: Use `\verbatim@font` instead of `\tt`. 208
- `\verbatim@font`: Macro added 208

1993/11/22 ltclass.dtx v0.2f
- `\@fileswithoptions`: Made the default [] not [`\@unknownversion`] 372
  - Made the initial version [] not [`\@unknownversion`] 373
- `\@ifclasslater`: Added //00 so parsing never produces a runaway argument. 367
- General: `\@unknownversion` removed 377

1993/11/22 ltdefns.dtx LaTeX2e
- `\@minus`: Macro added 23
- `\@plus`: Macro added 23
- `\CheckCommand`: Macro added 28
- `\providecommand`: Macro added 28

1993/11/22 lterror.dtx LaTeX2e
- `\c@errorcontextlines`: Macro added 42

1993/11/22 ltfiles.dtx LaTeX2e
- `\listfiles`: Removed checking for `\@unknownversion` 67

1993/11/22 ltlength.dtx LaTeX2e
- `\@settodim`: Macro added 106
- `\@settopoint`: Macro added 106
- `\settodepth`: Macro added 106
- `\settoheight`: Macro added 106

1993/11/22 ltlogos.dtx LaTeX2e
- `\LaTeXe`: Macro added 59

1993/11/23 ltclass.dtx v0.2g
- `\@use@ption`: Name changed from `\@executeoption` 370
- General: Various macros now moved to latex.tex. 365
  - Warnings and errors now directly coded. 365

1993/11/23 ltdefns.dtx LaTeX2e
- `\@argdef`: Macro added 25
- `\@ifundefined`: Redefined to remove a trailing `\fi` 32
- `\@newcommand`: Macro added 25
- `\@newenv`: Macro interface changed 27
- `\@xargdef`: Macro interface changed 25
- `\@yargd@f`: Avoid `\@?@?` token 26
  - Macro interface changed 26
- `\newcommand`: Macro reimplemented and extended 25
- `\renewcommand`: Macro reimplemented and extended 26
- `\renewenvironment`: Macro reimplemented and extended 27
- `\two@digits`: Macro added 22

1993/11/23 ltoutput.dtx v0.1a
- `\paperheight`: Register added 324
- `\paperwidth`: Register added 324

1993/11/23 ltoutput.dtx v0.1c
- `\@enlargepage`: Command added 350
- `\@kludgeins`: Insert added 350
- `\@makecol`: Command changed 332
- `\@specialoutput`: Command changed 328

`\enlargethispage*`: Commands added 350

1993/11/24 ltfntcmd.dtx v2.1a
- `\maybe@ic@`: Use `\t@st@ic` 195
- `\t@st@ic`: Macro added 195

1993/11/24 ltfssini.dtx v2.1a
- General: Removed `\xpt` stuff 170

1993/11/24 ltlogos.dtx LaTeX2e
- `\LaTeX`: Macro changed 59

1993/11/28 ltclass.dtx v0.2h
- `\@twoclasseserror`: Macro added 375
- General: Assorted commands now in the kernel removed. 365
  - Directory syntax checing moved to dircheck.dtx 365
  - Primitive filenames now terminated by space not `\relax`. 365
- `\endfilecontents`: Don't globally allocate a write stream (always use 15) 375

1993/11/28 ltfiles.dtx LaTeX2e
- `\@missingfileerror`: Use filename parser from dircheck 66

1993/11/29 ltoutput.dtx v1.0b
- `\@makecol`: `\@makespecialcolbox` added 332
- `\@makespecialcolbox`: Command added 333

1993/11/29 ltplain.dtx LaTeX2e
- General: All accents in decimals; suggested by Paul Taylor 19

1993/11/30 ltoutput.dtx v1.0c
- `\@tracemessage`: Commands added 351

1993/12/01 fontdef.dtx v2.1a
- General: Update for LaTeX2e 173

1993/12/01 ltoutput.dtx v1.0e
- `\@reinserts`: Command added 333

1993/12/03 ltboxes.dtx v0.1a
- `\@argrsbox`: macro removed 240
- `\@begin@tempboxa`: macro added 233
- `\@end@tempboxa`: macro added 234
- `\@iirsbox`: redefined to support `\height` 240
- `\@imakebox`: macro modified 234
- `\@irsbox`: redefined to support `\height` 240
- `\@isavebox`: color support 235
  - extra group 235
- `\@isavepicbox`: extra group 235
- `\@makebox`: default changed from x to c 233
- `\@makepicbox`: macro modified 234
- `\@savebox`: default c not x 235
- `\bm@b`: macros added 234
- `\endlrbox`: macro added 235
- `\fbox`: extra group 236
- `\lrbox`: color support 235
  - macro added 235
- `\makebox`: modified 233
- `\mbox`: extra group 233
- `\minipage`: Redefined to support extra optional arguments 238
- `\newsavebox`: Pass the whole of arg 1 to `\@ifdefinable` 235
- `\parbox`: Redefined to support extra optional arguments 237
- `\raisebox`: redefined to support `\height` 240
- `\sbox`: color support 235
  - extra group 235
- `\set@color`: color support 234
  - macro added 234

1993/12/03 ltclass.dtx v0.2i
- `\@cls@pkg`: Name changed to avoid clash with output routine. 374
- General: `\@onlypreamble`: Many commands declared. 365
  - Removed obsolete `\@documentclass` 365

1993/12/03 lterror.dtx v1.0b
- `\@latexerr`: Set `\c@errorcontextlines` to -1 42

1993/12/03 ltfssini.dtx v2.1a
- General: update for LaTeX2e 168

1993/12/04 ltfiles.dtx v0.9b
- `\@iinput`: Macro reimplemented 66
- `\@input`: Macro reimplemented 66
- `\IfFileExists`: Macro added 65
- `\input`: Macro reimplemented 66
- `\InputIfFileExists`: Macro added 66

1993/12/05 ltfloat.dtx LaTeX2e
- `\@dblfloatplacement`: Command changed 297
- `\@xfloat`: Command changed 294

1993/12/05 ltoutput.dtx v1.0f
- `\@addtobot`: Command changed 341
- `\@addtocurcol`: Command changed 342
- `\@addtodblcol`: Command changed 346
- `\@addtonextcol`: Command changed 345
- `\@addtotoporbot`: Command changed 342
- `\@boxfpsbit`: Command added 353
- `\@flcheckspace`: Command added 355
- `\@flsetnum`: Command added 354
- `\@flsettextmin`: Command added 354
- `\@flstop`: Commands added 352
- `\@flupdates`: Command added 356
- `\@fpsadddefault`: Command added 352
- `\@getfpsbit`: Command added 353
- `\@opcol`: Command changed 331
  - Hook added 331
- `\@outputpage`: Command changed 335
- `\@resethfps`: Command added 354
- `\@setfloattypecounts`: Command added 353
- `\@setfpsbit`: Command added 353
- `\@shipoutsetup`: Command added 335

- `\@startcolumn`: Command changed 338
- `\@startdblcolumn`: Command changed 338
- `\@testfp`: Command added 353
- `\@textfloatsheight`: Commands added 352
- `\@topnewpage`: Commands changed 326
- `\@tryfcolumn`: Command changed 339
- `\@writesetup`: `\@startpagehook` added 335
- `\output`: Command changed 328

1993/12/06 ltclass.dtx v0.2k
- `\ExecuteOptions`: Preserve `\CurrentOption`. 370

1993/12/06 ltoutput.dtx v1.0f
- `\@specialoutput`: Unboxing of 255 added to rescue writes 328

1993/12/06 ltoutput.dtx v1.0g
- `\@topnewpage`: `\@floatplacement` placement bug fixed 326

1993/12/07 ltclass.dtx v0.2l
- `\ProvidesFile`: Macro added 368

1993/12/07 ltclass.dtx v0.2m
- `\@fileswithoptions`: Reset `\CurrentOption` 373

1993/12/07 ltoutenc.dtx 1.1
- General: Protected all special characters with `\string`. 72

1993/12/07 ltoutenc.dtx v1.1
- General: Made all character numbers decimal. 69
  - Removed a lot of equal signs and the like. 69

1993/12/08 ltboxes.dtx v0.1b
- `\@begin@tempboxa`: Extra braces for color support (braces removed from other macros) 233
- `\@irsbox`: fix typo 240
- `\@parboxto`: `\endgraf` added due to extra group in `\@begin@tempboxa` 237
- `\lrbox`: move `\@endpefalse` out of the inner group 235

1993/12/08 ltfntcmd.dtx v2.1b
- General: Macros `\rm`, `\bf` and `\sf` moved to classes.dtx 197

1993/12/08 ltlists.dtx LaTeX2e
- `\@item`: use `\sbox` to support colour 229

1993/12/08 ltspace.dtx LaTeX2e
- `\@bsphack`: Command reimplemented 52
  - Command reimplemented; late birthday present for Chris 52
- `\@vbsphack`: Command added 54

1993/12/09 ltboxes.dtx v0.1c
- `\@irsbox`: fix another typo 240

1993/12/09 ltclass.dtx v0.2n
- `\documentstyle`: input 209 compatibility file. 370

1993/12/09 ltfiles.dtx v0.9e
- `\document`: Hook added 62

1993/12/09 ltmiscen.dtx v0.9e
- `\enddocument`: Hook added 203

1993/12/10 ltoutenc.dtx v1.2
- General: Added source code for t1enc.sty. 69

1993/12/11 ltfntcmd.dtx v3.0a
- General: Complete reworking of all text commands, using just one creator function 191
  - italic correction now put in front of penalty before glue 191
  - newcommands replaced by defs 191
  - newfontswitch command corrected and changed 191
- `\DeclareTextFontCommand`: Macro changed 193
- `\emph`: Macro changed 194
- `\fix@penalty`: Macro added 196
- `\maybe@ic`: Macro name changed 195
- `\maybe@ic@`: Macro and name changed 195
- `\sw@slant`: Macro changed 196
- `\textup`: Macros changed 193

1993/12/11 ltmath.dtx v0.9g
- General: Added a group around the first argument of `\frac` to prevent changes (for example font changes) from modifying the contents of the second argument. 215

1993/12/11 ltoutenc.dtx v1.2a
- General: Corrected for t1enc, math. 69

1993/12/11 ltsect.dtx LaTeX2e
- `\@author`: Added default 282
- `\@title`: Added default 282

1993/12/11 ltxref.dtx LaTeX2e
- `\@setref`: Macro added 200
- `\pageref`: Macro reimplemented 200
- `\ref`: Macro reimplemented 200

1993/12/12 ltoutput.dtx v1.0h
- `\@cflb`: boxmaxdepth setting moved 337
  - defs changed to lets 337
- `\@cflt`: name changed 337
- `\@doclearpage`: defs changed to lets 331
- `\@makecol`: defs changed to lets 332
- `\@resethfps`: Warnings added: minimal 354
- `\@startdblcolumn`: defs changed to lets 339
- `\@topnewpage`: braces removed 326
- `\@tracemessage`: Commands changed 351
- `\@tryfcolumn`: defs changed to lets 339

1993/12/13 ltclass.dtx v0.2o
- General: Removed setting `\errorcontextlines` (now in latex.tex) 365

- `\documentstyle`: compatibility file now latex209.sty. 370
- `\usepackage`: Fixed error handling 371

1993/12/13 ltdirchk.dtx v0.2a
- General: on the 'docstrip' pass, do not check openin path 8
- `\IfFileExists`: Removed interactive prompting for current directory syntax 7
- `\strip@prefix`: modified, name changed from `\stripmeaning`. 3

1993/12/13 ltlists.dtx latex2e
- `\trivlist`: Initialised `\@itemlabel` 226

1993/12/13 ltmiscen.dtx v0.9h
- `\@noligs`: Readded `\@noligs` 209
- `\@verbatim`: Readded `\@noligs` 207
  - Removed optional argument of `\item` 207
- `center`: Removed optional argument of `\item` 206
- `flushleft`: Removed optional argument of `\item` 206
- `flushright`: Removed optional argument of `\item` 206

1993/12/13 ltoutenc.dtx v1.2b
- General: Corrected file name in driver code. 69

1993/12/13 lttab.dtx latex2e
- `\tabbing`: Removed optional argument of `\item` 246

1993/12/14 ltoutput.dtx v1.0i
- General: Section added to declare all parameters 359

1993/12/15 ltboxes.dtx v0.1d
- `\@iminipage`: Changed default from 'c' to 's' 238
- `\@iparbox`: Changed default from 'c' to 's' 237
- `\minipage`: Changed default from 'c' to 's' 238
  - extra space removed. 238
- `\parbox`: Changed default from 'c' to 's' 237

1993/12/15 ltclass.dtx v0.2p
- General: Removed extra '.'s from `\@@warnings` 365

1993/12/16 ltlogos.dtx LaTeX2e
- `\LaTeXe`: Extended logo by DPC 59

1993/12/16 ltmath.dtx v0.9i
- `\@@eqncr`: use `\refstepcounter` instead of shortcut 216
- General: use `\refstepcounter` instead of shortcut 215

1993/12/16 ltmiscen.dtx v0.9i
- General: `\literal` added 209

1993/12/16 ltpage.dtx LaTeX2e
- `\mark`: Init `\mark` at begin document 312

1993/12/16 ltspace.dtx LaTeX2e
- `\@bsphack`: Corrected optimisation :-) 52

1993/12/16 lttab.dtx latex2e
- `\@xhline`: Measure from middle of vertical rules 258

1993/12/17 ltclass.dtx v0.2q
- `\@documentclasshook`: Macro added 365
- `\@fileswithoptions`: Add `\@compatibility` hook 372
- `\documentstyle`: Match Alan's new code. 370

1993/12/17 ltoutenc.dtx 1.3
- General: Added this section 72
  - Removed all the hackery for use in `\DeclareFontEncoding`, and redid everything using `\DeclareTextFoo`. 81, 82
  - 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. 72

1993/12/17 ltoutenc.dtx v1.3
- General: Added `\EncodingSpecificAccent`, `\EncodingSpecificAccentedLetter` and `\EncodingSpecificCommand`. 69
  - Made Rokicki's encoding a proper encoding scheme rather than a variant of OT1. 69

1993/12/17 ltoutput.dtx v1.0j
- `\@opcol`: Hook removed 331
- `\@specialoutput`: Page room test added 328
- `\@topnewpage`: check for vsize too small added 326
  - Page room test added 327
- `\@tracemessage`: tracefloatvals made a document command 351
- `\@writesetup`: —and then removed 335

1993/12/17 ltpage.dtx LaTeX2e
- `\mark`: Removed init `\mark` at begin document, since it doesn't work. 312
- `\rightmark`: Stopgap solution to mark `\leftmark` and `\rightmark` work without initializing mark until the problem is solved. 312

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. 72

Replaced the missing last argument to `\DeclareFontEncoding`. 81, 82

1993/12/18 ltoutenc.dtx 1.3c
- General: Rewrote for the new syntax of `\EncodingSpecific`. 81, 82
  - Split `\EncodingSpecificAccent` up into `\EncodingSpecific` and `\DeclareAccent`. 72

1993/12/18 ltoutenc.dtx v1.3a
- General: Replaced OT3 by XXX 69

1993/12/18 ltoutenc.dtx v1.3b
- General: Corrected typos. 69
  - Replaced the missing last argument to `\DeclareFontEncoding`. 69

1993/12/18 ltoutenc.dtx v1.3c
- General: A new syntax, separating accent-definitions from encoing-specific definitions, and allowing encoding-specific `\chardef`, `\let`, etc. 69
  - Rewrote for the new syntax of `\EncodingSpecific`. 69

1993/12/18 ltoutenc.dtx v1.3d
- General: Some T1 stuff had drifted into the OT1 file. 69

1993/12/18 ltpage.dtx LaTeX2e
- `\sloppy`: Added `\emergencystretch` 313

1993/12/19 ltclass.dtx v0.2r
- `\endfilecontents`: Different message when ignoring a file 375

1993/12/19 ltfntcmd.dtx v3.0b
- General: `\@pdef` comand added 191
  - Added by ASAJ. 197
  - Made `\@newfontswitch` produce an error if command already exists, and added `\@renewfontswitch`, ASAJ 191
  - Other tidying 191
  - Some more tidying done 191
  - Untidying added, so this is now a TEMPORARY version. 191
  - Wording changes by CAR. 197
- `\DeclareOldFontCommand`: Corrected and tidied 197
- `\DeclareTextFontCommand`: Corrected and tidied 193

1993/12/19 ltspace.dtx LaTeX2e
- `\@bsphack`: There seem to be problems with selfmade birthday presents 53

1993/12/20 ltdefns.dtx LaTeX2e
- `\@reargdef`: Kept old version of `\@reargdef`, for array.sty 26

1993/12/20 ltfiles.dtx v0.9m
- `\@obsoletefile`: Added this command, removed @oldfilewarning 67

1994/01/05 fontdef.dtx v2.1d
- General: Removed nf prefix from file names. 175

1994/01/13 ltmath.dtx v0.9o
- `\@@eqncr`: correcting 0.9i 216
- General: correcting 0.9i 215

1994/01/14 ltdirchk.dtx v0.2d
- `\IfFileExists`: Close the texsys.aux output stream 8

1994/01/15 ltfiles.dtx v0.9o
- `\document`: move `\@preamblecmds` after document hook 63

1994/01/17 ltclass.dtx v0.2s
- `\@fileswithoptions`: Modify to reduce parameter stack usage 372, 373
- General: Added many more `\@onlypreamble` commands 365
  - Wrapped long lines to column 72 365

1994/01/17 ltfiles.dtx LaTeX2e
- `\listfiles`: New Version, adds '.tex' if needed, and lines up columns 67

1994/01/17 ltfssbas.dtx v2.1a
- General: New math font setup 107
- `\curr@math@size`: New math font setup 115
- `\everydisplay`: New math font setup 115
- `\everymath`: New math font setup 115
- `\frozen@everydisplay`: New math font setup 115
- `\frozen@everymath`: New math font setup 115
- `\math@version`: New math font setup 114

1994/01/17 ltfssini.dtx v2.1e
- `\not@math@alphabet`: Message changed 169

1994/01/17 ltfsstrc.dtx v2.3a
- General: New math font setup 125
- `\check@mathfonts`: New math font setup 132
- `\glb@currsize`: New math font setup 130
- `\restglb@settings`: New math font setup 133

1994/01/18 ltbibl.dtx LaTeX2e
- `\bibliography`: Use `\@input@` so include files are listed. 309

1994/01/18 ltclass.dtx v0.2t
- `\@ifclassloaded`: Fix typo `\@pkgetension` 367

1994/01/18 ltfiles.dtx v0.9p
- `\@iffileonpath`: Macro added 65
- `\@input`: do not use a different definition for `\input@path` 66
- `\@input@`: Macro added 66
- `\IfFileExists`: New Definition 65
- `\include`: Use `\@input@` so include files are listed. 64

- \InputIfFileExists: New Definition . . . 66

1994/01/18 ltfssini.dtx v2.1f
- \not@math@alphabet: Message corrected . . . 169

1994/01/18 ltmiscen.dtx v0.9p
- \@verbatim: Add \global\@inlabelfalse . . . 207
  - Only add \penalty if in hmode 207

1994/01/19 fontdef.dtx v2.1e
- General: Added missing setting for symbols in bold version. . . . 177

1994/01/19 ltdirchk.dtx v0.2e
- \IfFileExists: name changed from \test . . . 7
- \input@path: No longer check that an empty group is in the path . 8
- \strip@prefix: name changed from \strip@meaning, to match NFSS. . . . 3

1994/01/19 ltmath.dtx v1.0n classes
- \mathindent: Deferred setting of \mathindent . . . 217

1994/01/20 ltdirchk.dtx v0.2f
- General: \@copytexsys and the texsys.new file removed . . . 7
  - Modify all of ltxcheck . . . 10
- \IfFileExists: \@copytexsys removed . . . 7

1994/01/21 ltclass.dtx v0.2u
- \documentstyle: compatibility file now latex209.def. . . . 370

1994/01/21 ltdirchk.dtx v0.2g
- General: Improve documentation, reorganise docstrip module . . . 1
- \filename@parse: Minor changes, and add Mac version (:) . . . 9
- \today: Name changed from \stamp, to save memory . . . 7

1994/01/21 ltfloat.dtx LaTeX2e
- \@xfloat: Added missing percent characters. . . . 294

1994/01/21 ltmiscen.dtx v0.9s
- \verbatim@font: Removed unnecessary category code hackery. 208

1994/01/24 ltdirchk.dtx v0.2h
- \IfFileExists: Stop testing once texsys.aux has been found . . . 7

1994/01/24 ltpage.dtx LaTeX2e
- \pagestyle: (DPC) Complain if pagestyle is undefined. . . . 311

1994/01/25 ltdirchk.dtx v0.2i
- General: Protect against looping on \@@input and \@@end. . . . 2

1994/01/25 ltfssbas.dtx v2.1b
- \math@version: Corrections for math setup . . . 114

1994/01/25 ltmath.dtx LaTeX2e
- \bordermatrix: Removed \p@renwd. . . . 212

1994/01/26 ltfsstrc.dtx v2.3c
- \check@mathfonts: Correct trace info placement . . . 132
- \restglb@settings: Correct trace info placement . . . 133

1994/01/27 ltfntcmd.dtx v3.1a
- \nocorrlist: Only ., used as default for cm fonts . . . 196

1994/01/29 ltclass.dtx v0.2v
- \@@unprocessedoptions: Macro added. . . . 375
- \@fileswithoptions: All options raise error if no \ProcessOptions appears . . . 373

1994/01/31 ltclass.dtx v0.2w
- \g@addto@macro: Use toks register to avoid ‘hash’ problems . . . 374

1994/01/31 ltfiles.dtx v0.9t
- \document: set \@normalsize or \normalsize if necessary . . . 62

1994/01/31 ltfntcmd.dtx v3.1b
- General: \@normalsize no longer defined . . . 191

1994/02/01 ltpage.dtx LaTeX2e
- \pagestyle: (DPC) Modify to get nicer error message . . . 311
- \thispagestyle: (DPC) Modify to get nicer error message . . . 311

1994/02/02 ltclass.dtx v0.2x
- \@fileswithoptions: Only run the hook and options check if the file was loaded. . . . 373

1994/02/03 ltoutput.dtx v1.0k
- \@makespecialcolbox: correct mistakes in the documentation . 334

1994/02/07 ltclass.dtx v0.2y
- \@fileswithoptions: Run \@compatibility on the first class to start (not the first to finish) 372
- \@ifclasswith: Add extra ,s so ‘two’ is not matched with ‘twocolumn’ . . . 367
- \ProcessOptions*: Add extra ,s so ‘two’ is not matched with ‘twocolumn’ . . . 369

1994/02/07 ltfssbas.dtx v2.1c
- \DeclareFontEncoding: revert catcode settings earlier . . . 109
- \DeclareFontShape: revert catcode settings earlier . . . 108

1994/02/08 ltoutput.dtx v1.0k
- \@makespecialcolbox: boxmaxdepth setting added . . . 334
  - boxmaxdepth setting removed 333
- General: Documentation and tasks tidied. . . . 314

1994/02/10 ltclass.dtx v0.2z
- \@documentclasshook: Changed the name from \@compatibility to \@documentclasshook, and added the check for

whether `\@normalsize` has been defined. ASAJ. 365

- `\@fileswithoptions`: Renamed `\@compatibility` to `\@documentclasshook`. ASAJ. 372

1994/02/10 ltfssbas.dtx v2.1d
- `\addto@hook`: Made `\addto@hook` long. 123

1994/02/10 ltfsscmp.dtx v2.1d
- `\scan@@fontshape`: scan away stuff after pt 146

1994/02/22 ltfssini.dtx v2.1g
- General: Correct error mssage . . 171

1994/02/24 ltfssbas.dtx v2.1e
- `\DeclareFontShape`: Separate restoration of catcodes for fd cmds 108
- `\define@newfont`: Separate restoration of catcodes for fd cmds 117
- `\nfss@catcodes`: Separate restoration of catcodes for fd cmds . 118

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 2ekernel module . . 233
  - Remove need for drv file 233

1994/03/02 ltclass.dtx v0.3a
- General: Remove need for driver file 365

1994/03/03 ltboxes.dtx v0.1f
- `\@irsbox`: Replaced a missing `\else` 240

1994/03/04 ltfloat.dtx v1.0a
- General: Initial version, split from latex.dtx 291

1994/03/04 ltsect.dtx v1.0a
- General: Initial version, split from latex.dtx 282

1994/03/04 lttab.dtx v1.0a
- General: Initial version, split from latex.dtx 241

1994/03/04 ltvers.dtx v1.0a
- General: Initial version, split from latex.dtx 21

1994/03/07 ltboxes.dtx v0.1a
- `\@mpfootnotetext`: Extra group for colour 239

1994/03/07 ltboxes.dtx v1.0a
- General: Unify format with other Kernel files 233

1994/03/07 ltdefns.dtx v1.0a
- `\@@italiccorr`: Macro added . . . 23

1994/03/07 ltfiles.dtx v1.0a
- General: Initial version, split from latex.dtx 60
  - Long lines wrapped to 72 columns 60

1994/03/07 ltfinal.dtx v0.1a
- General: Add code from the old dump.dtx 383
  - Initial version, split from latex.dtx 379
  - move code here from lhyphen.dtx 380
  - Remove oldcomments environment 379
  - use `\InputIfFileExists` not `\IfFileExists` 380

1994/03/07 ltfloat.dtx v1.0a
- `\@endfloatbox`: (DPC) Extra group for colour 297
- `\@footnotetext`: (DPC) Extra group for colour 304
- `\@xfloat`: (DPC) Extra group for colour 295

1994/03/07 lthyphen.dtx v0.1c
- General: move the 2ekernel code to ltfinal.dtx 378

1994/03/07 ltlength.dtx v1.0a
- `\@settodim`: (DPC) Extra group for colour 106

1994/03/07 ltlists.dtx v1.0a
- General: Initial version, split from latex.dtx 219
  - Long lines wrapped to 72 columns 219

1994/03/07 ltpage.dtx v1.0a
- General: Initial version, split from ltherest.dtx 311

1994/03/07 ltpictur.dtx v0.1a
- General: Initial version, split from latex.dtx 260
  - Long lines wrapped to 72 columns 260

1994/03/07 ltsect.dtx v1.0a
- `\@hangfrom`: (DPC)Extra groups for colour 287

1994/03/07 lttab.dtx v1.0a
- General: Long lines wrapped to 72 columns 241

1994/03/08 ltclass.dtx v0.3b
- General: Modify driver code into ‘new style’ 365

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. 11

1994/03/10 ltfssbas.dtx v2.2f
- `\math@egroup`: Changed `\begingroup`/`\endgroup` to `\bgroup`/`\egroup`. 122

1994/03/11 ltfssdcl.dtx v2.1b
- `\DeclareSymbolFontAlphabet@`: Added check against use of alphabet switch outside of math mode. 167

- `\SetMathAlphabet@`: Changed parameter template in temporary macro to catch check add below. 159

1994/03/12 ltclass.dtx v0.3c
- `\@fileswithoptions`: Do not use `\@pr@videpackage` to avoid typeout 373
- General: Change name from docclass to ltclass 365
- `\ProvidesFile`: Add `\wlog` 368
- `\ProvidesPackage`: Add `\wlog` 367
  - use `\@gtempa` 367

1994/03/12 ltdefns.dtx v1.0b
- `\@reargdef`: New defn, in terms of `\@yargdef` 26
- `\@yargd@f`: Name changed from `\XXX@argdef` 26

1994/03/12 ltdirchk.dtx v1.0b
- General: Change name from dircheck.dtx 1
  - Minor edits to the typeouts in ltxcheck 1

1994/03/12 ltfloat.dtx v1.0b
- `\@savemarbox`: (DPC) Extra group for colour 299
- `\@xympar`: (DPC) Extra bgroup for colour 300

1994/03/12 ltplain.dtx v1.0b
- General: Name changed from lplain. The end of an era 11

1994/03/12 ltplain.dtx v1.0e
- General: Replaced remaining width, height, depth by LaTeX macro names to save tokens. 11

1994/03/13 ltcntrl.dtx v1.0c
- `\@tfor`: (DPC) Add `\@tf@r` so a single group is correctly treated. 37

1994/03/13 ltfiles.dtx LaTeX2e
- `\@addtofilelist`: Macro added 67
- `\listfiles`: Reset `\@addtofilelist` at begin document 67

1994/03/13 ltfiles.dtx v0.3b
- `\InputIfFileExists`: Use new cmd `\@addtofilelist` 66

1994/03/13 ltfssbas.dtx v2.1g
- General: add 2ekernel module to omit repeated code 107

1994/03/13 ltfssdcl.dtx v2.1c
- General: add 2ekernel module to omit repeated code 149

1994/03/14 ltboxes.dtx v1.0b
- `\@isavebox`: Use `\color@setgroup` 235
- `\@isavepicbox`: Use `\color@setgroup` 235
- `\color@begingroup`: macro added for colour support 234
- `\color@endgroup`: macro added for colour support 234
- `\lrbox`: Use `\color@setgroup` 235
- `\sbox`: Use `\color@setgroup` 235

1994/03/14 ltfloat.dtx 1.0c
- `\@xympar`: (DPC) Use `\color@begingroup` 300

1994/03/14 ltfloat.dtx v1.0c
- `\@endfloatbox`: (DPC) Use `\color@endgroup` 297
- `\@footnotetext`: (DPC) Use `\color@begingroup`, add `\endgraf` 304
- `\@savemarbox`: (DPC) Use `\color@begingroup` 299
- `\@xfloat`: (DPC) Use `\color@begingroup` 295

1994/03/15 ltfiles.dtx LaTeX2e
- `\@missingfileerror`: Quit on x or X just like a real error 66

1994/03/15 ltfntcmd.dtx v3.2a
- General: Adapted to mass formatting 191
  - Changed `\/` to `\@@italiccorr` 191
  - Removed `\@renewfontswitch` 191
  - Removed defs of short-forms and all sizes except `\normalize` 191

1994/03/15 ltoutput.dtx v1.0l
- `\@addtocurcol`: Changed `\addvspace` to `\vskip` 344
- `\@combinedblfloats`: Removed boxmaxdepth setting. 338
- `\@makecol`: `\maxdepth` changed to `\@maxdepth` 332
  - Removed boxmaxdepth setting. 333
- `\@makespecialcolbox`: Removed boxmaxdepth setting. 334
- `\@topnewpage`: Corrected and amended warning message 327
  - Warning added: it should be improved 327
- General: Added some warnings when page gets full of top floats. 314
  - Driver added and further tidying. 314
  - Removed duplicated code and corrected docstrip options. 314
  - Some boxmaxdepth settings removed. 314

1994/03/16 ltclass.dtx v0.3f
- General: Add pkgindoc package 377

1994/03/16 ltfiles.dtx LaTeX2e
- `\listfiles`: Move this code directly into `\document` 67

1994/03/16 ltfiles.dtx v1.0c
- `\document`: (DPC) directly add file list settings 63

1994/03/16 ltmiscen.dtx v1.0b
- `\@verbatim`: Remove `\global\@inlabelfalse` again. 207

1994/03/28 ltalloc.dtx v1.0d
- General: Redefinition of 'new' allocations removed. 34

1994/03/28 ltdirchk.dtx v1.0d
- General: Improve documentation . 1

1994/03/28 lterror.dtx v1.0d
- `\@invalidchar`: (DPC) Comment out (use catcode15 instead) . . 45
- General: Remove test for `\inputlineno` undefined. 42

1994/03/28 ltfiles.dtx v1.0d
- `\document`: (DPC) Use `\normalsize` not `\@normalsize` 62
  - (DPC) remove `\@normalsize check` 62

1994/03/28 ltfloat.dtx v1.0b
- `\@caption`: Use `\normalsize` not `\@normalsize` 293
- General: Split further from ltherest.dtx 291

1994/03/28 ltlists.dtx v1.0b
- General: Improve documentation 219

1994/03/28 ltmiscen.dtx v1.0c
- General: Improve Documentation 202

1994/03/28 ltplain.dtx v1.0c
- `\newtoks`: Remove some `\outer` declarations. 13

1994/03/28 ltsect.dtx v1.0b
- General: Split further from ltherest.dtx 282

1994/03/28 lttab.dtx v1.0b
- General: Improve documentation 241

1994/03/28 ltthm.dtx v1.0a
- General: Initial version, split from latex.dtx 279

1994/03/29 ltcounts.dtx v1.0c
- General: Create file from parts of ltmiscen and ltherest. 103

1994/03/29 ltlength.dtx v1.0c
- General: Create file ltcntlen from parts of ltmiscen and ltherest. 106

1994/03/29 ltmiscen.dtx v1.0d
- General: Remove counter macros to ltcntlen 202

1994/03/29 ltpageno.dtx v1.0c
- General: Create file ltcntlen from parts of ltmiscen and ltherest. 198

1994/03/29 ltxref.dtx v1.0c
- General: Create file ltcntlen from parts of ltmiscen and ltherest. 199

1994/03/31 ltbibl.dtx v1.0a
- General: Initial version of ltidxbib.dtx, split from ltherest.dtx 307

1994/03/31 ltidxglo.dtx v1.0a
- General: Initial version of ltidxbib.dtx, split from ltherest.dtx 305

1994/04/09 ltcounts.dtx v1.0d
- `\@newctr`: `\@nocnterr` now has counter name argument 104
- `\addtocounter`: `\@nocnterr` now has counter name argument . 104
- `\setcounter`: `\@nocnterr` now has counter name argument 104
- `\stepcounter`: Use `\addtocounter` to have name checked 104

1994/04/09 ltthm.dtx v1.0b
- `\@othm`: Use standard counter error message (FMi) 281

1994/04/11 ltclass.dtx v0.3g
- `\endfilecontents`: Add star form, dont write `\endinput` at the end of the file. 375
- `\ProvidesFile`: Protect against weird catcodes. 368

1994/04/11 ltfssbas.dtx v2.1h
- General: Added `\defaultscriptratio` and `\defaultscriptscriptratio`. ASAJ. . 107
- `\defaultscriptratio`: Macro added 122
- `\defaultscriptscriptratio`: Macro added 122

1994/04/12 ltboxes.dtx v1.0c
- General: Remove `\@acci`, now defined in ltplain.dtx 237
  - Remove `\@dischyph`, now defined in ltinit.dtx 237

1994/04/12 ltdefns.dtx v1.0g
- `\@dischyph`: Define `\@dischyph`, was previously in ltboxes.dtx . 23

1994/04/12 ltplain.dtx v1.0d
- General: Define `\@acci` 19

1994/04/12 ltvers.dtx v1.0b
- General: Have version info generated automatically. 21

1994/04/14 ltfntcmd.dtx v3.2b
- General: Macros renamed to non-private forms, JB 191
- `\DeclareOldFontCommand`: Renamed from `\@newfontswitch` 196

1994/04/15 ltboxes.dtx v1.0d
- `\@isavebox`: Added missing precent character. 235

1994/04/17 ltcounts.dtx v1.0e
- `\@newctr`: Use `\@nocounterr` instead of `\@nocnterr` 104
- `\addtocounter`: Use `\@nocounterr` instead of `\@nocnterr` 104
- `\setcounter`: Use `\@nocounterr` instead of `\@nocnterr` 104

1994/04/17 lterror.dtx v1.0h
- `\@nocounterr`: New name for error message, old error message (without arg) kept 43

1994/04/17 ltthm.dtx v1.0c
- `\@othm`: Use new std counter error message (FMi) 281

1994/04/18 ltfinal.dtx v0.1b
- General: Initialise `\textheight`, `\textwidth` and page style . 379

1994/04/18 ltfloat.dtx v1.0d
- `\@footnotetext`: (DPC) Remove Colour support 304
- `\@savemarbox`: (DPC) Remove Colour support 299

- 1994/04/18 ltfssbas.dtx v2.1i
  - General: Macro `\no@alphabet@help` removed again 107
  - `\calculate@math@sizes`: Changed message to log only 122
  - `\no@alphabet@error`: Use std LaTeX error macro 107
- 1994/04/18 ltfssdcl.dtx ???
  - `\DeclareMathAlphabet`: Pass correct arg (2 not 3) 157
- 1994/04/18 ltfssdcl.dtx v2.1d
  - General: Removed surplus `\no@alphabet@error` (see fam.dtx) 149
- 1994/04/18 ltfsstrc.dtx v2.3d
  - General: Changed to new error/warning scheme 125
  - `\font@submax`: Changed dimen to macro 140
  - `\fontsubfuzz`: Changed dimen to macro 140
  - `\subst@size`: `\font@submax` and `\fontsubfuzz` now macros 141
- 1994/04/19 ltpage.dtx v1.0b
  - General: Improve documentation 311
- 1994/04/20 ltfntcmd.dtx v3.3a
  - General: Documentation up-dated 191
    - New implementation of `\nocorr` 191
  - `\check@nocorr@`: Macros added 194
  - `\maybe@ic@`: `\nocorr` etc removed from list of tokens to check, leaving only punctuation characters 195
- 1994/04/20 ltmiscen.dtx v1.0e
  - `\enddocument`: Changed logic for producing warning messages 204
- 1994/04/21 ltboxes.dtx v1.0e
  - `\@iiiminipage`: Extra `\bgroup` for colour 238
  - `\@mpfootnotetext`: Extra `\endgraf` for colour 239
  - `\endminipage`: Extra `\egroup` for colour 239
- 1994/04/21 ltfinal.dtx v0.1c
  - General: Added comments, set the catcodes of 128–255. 379
- 1994/04/22 ltfssini.dtx v2.1g
  - `\not@math@alphabet`: Message changed again 169
- 1994/04/23 ltfinal.dtx v0.1d
  - General: Check that `\font@submax` is still zero 379
- 1994/04/24 ltoutput.dtx v1.0m
  - `\@resethfps`: Number 2 changed to `\tw@` 354
    - Warning changed 354
  - `\@specialoutput`: Message changed to give more info and 'top' removed 328
  - `\@topnewpage`: Message changed to give more info 327
    - Warning message removed as it will be generated later 327
  - General: Changed `\@normalsize` to `\normalsize`. 314
    - Corrected unverbed commands in documentation. 314
    - Removed some long lines and other aesthetic changes. 314
    - Warning messages changed/corrected. 314
- 1994/04/24 ltpictur.dtx v0.1b
  - General: Removed surplus spaces after `\hbox to` in several cases 260
- 1994/04/25 ltclass.dtx v0.3h
  - General: Removed spurious extra '.'s at the end of error messages 365
- 1994/04/25 ltfloat.dtx v1.0e
  - `\@largefloatcheck`: Changed warning message to give more info 297
    - Command added 297
  - General: Changed warning messages 291
    - Removed obsolete tracing code 291
- 1994/04/27 ltfsstrc.dtx v2.3e
  - General: Corrected item that was forgotten in last change. 125
- 1994/04/28 lterror.dtx v1.0j
  - `\@inmatherr`: Macro added 45
- 1994/04/28 lterror.dtx v1.1c
  - `\@inmatherr`: Replaced `\noexpand` with `\protect`. 45
- 1994/04/28 ltfssdcl.dtx v2.1e
  - General: Removed all `\uppercase` in hex num parsing macros 149
- 1994/04/28 ltlists.dtx v1.0c
  - General: Replaced `\@ltxnomath` by `\@inmatherr` 227
- 1994/04/28 ltpictur.dtx v0.1c
  - General: bezier curves added 276
  - `\multiput`: (DPC) Ignore spaces between)(262
    - (DPC) Macro added 262
  - `\picture`: (DPC) Ignore spaces before (261
- 1994/04/28 ltplain.dtx v1.0g
  - General: Turn off overfull box tracing in log 16
- 1994/04/29 ltclass.dtx v1.0a
  - General: Change version number to 1 (no other change) 365
- 1994/04/29 ltmiscen.dtx v1.0f
  - `\@verbatim`: `\leavevmode` added 207
    - Change to `\everypar` added 208
- 1994/04/29 ltoutenc.dtx 1.4a
  - General: Removed `\EncodingSpecific`. Renamed all the commands. Added `\DeclareTextGlyph` and `\UndeclareTextCommand`. 72

- Removed Rokicki's OT1 variant encoding. Moved the driver to the top. 72

1994/04/30 ltfntcmd.dtx v3.3b
- General: Documentation up-dated and tidied 191
  - Prefix frag@ changed to frag in `\@protecteddef` 191
  - Title changed 191
  - Warning changed to info message in `\@protecteddef` 191

1994/04/30 ltoutput.dtx v1.0n
- `\@activechar@info`: `\@activechar@warning` changed to `\@activechar@info` 335
- `\@combinedblfloats`: Removed rule in topnewpage case 338
- `\@emptycol`: Empty column action added: `\@emptycol` 326
- `\@flsetnum`: Rogue space removed 354
- `\@specialoutput`: Cut-off point changed to 2`\baselineskip` 328
  - Empty column action added: `\@emptycol` 328
  - Extra empty column added for twocolumn case 328
  - Extra empty column added for twocolumn case (wrong, see below) 328
- `\@topnewpage`: Added setting of `\col@number` 326
  - Cut-off point changed to 3`\baselineskip` 327
  - Empty column action added: `\@emptycol` 327
  - Message changed for Frank 327
- General: `\@activechar@warning` changed to an info message. 314
  - Added `\col@number`. 314
  - Documentation tidied. 314
  - Empty column action added. 314
  - Fixed bug from `\dblfigrule` with `\@topnewpage`. 314
  - Full of floats action improved. 314
- `\col@number`: Added `\col@number` 324
- `\onecolumn`: Added setting of `\col@number` 326

1994/05/01 lterror.dtx v1.0k
- `\@latexerr`: (CAR) Added draft `\@latexinfo`. 42

1994/05/01 ltoutenc.dtx 1.4a
- General: Added the `\a` command. 78
  - Added the `\SaveAtCatcode` and `\RestoreAtCatcode` commands. 81
  - Removed the uc/lc table settings, since the T1 uc/lc table is now the default. 86
  - Rewrote for the new syntax. 81, 82

1994/05/01 ltoutenc.dtx v1.4a
- General: Removed Rokicki's encoding. 69
  - Renamed the commands, removed the `\EncodingSpecific` command. Turned all slots into decimal. Added `\a`. 69

1994/05/02 ltcntrl.dtx v1.0l
- `\@break@tfor`: Macro added (from ltfiles.dtx) 37

1994/05/02 ltfiles.dtx v1.0f
- `\@iffileonpath`: `\@break@loop` renamed to `\@break@tfor` 65
- `\@obsoletefile`: Make `\@onlypreamble` 67

1994/05/02 ltfinal.dtx v0.1e
- General: Added setting the 'letter' catcodes. 382
  - Added setting the 'other' catcodes. 382
  - Added setting the special catcodes. 381
  - Made slot 127 illegal 382
  - Set all the catcodes 379

1994/05/02 ltfinal.dtx v0.1f
- General: Set the catcode of control-J. 381

1994/05/02 ltmiscen.dtx v1.0g
- General: Changed 91 to 1991 and moved some bits 202

1994/05/02 ltoutput.dtx v1.0o
- `\@resethfps`: Code shortened 354
- General: Code of `\@resethfps` shortened. 314

1994/05/03 ltbibl.dtx v1.0b
- `\nocite`: Make `\nocite` issue a warning for an undefined citation key. 309

1994/05/03 ltfinal.dtx v0.1f
- General: Set the catcode of control-J to be 'other', for use in messages. 379

1994/05/03 ltfloat.dtx v1.0f
- General: (CAR) Added `\@largefloatcheck` 291
  - Removed unnecessary braces from arguments of `\@ifnextchar` 291
- `\end@dblfloat`: `\@largefloatcheck` added 296
- `\end@float`: (CAR) Added `\@largefloatcheck` 296

1994/05/03 ltfssdcl.dtx v2.1f
- General: Renamed `\@@DeclareMathDelimiter` to `\@DeclareMathDelimiter` 149

1994/05/03 ltlists.dtx v1.0d
- `\@item`: `\hskip` changed to `\kern` 228
- General: Removed superfluous braces 227

1994/05/03 ltmiscen.dtx v1.0h
- `\@centercr`: `\@badcrerr` replaced by `\@nolnerr` 206

1994/05/03 lttab.dtx v1.0d
- `\@endpbox`: Use `\@finalstrut` based on depth of `\@arstrutbox` 259

1994/05/04 ltclass.dtx v1.0b
- `\NeedsTeXFormat`: Changed wording of the warning 372

1994/05/04 lterror.dtx v1.0m
- `\@badcrerr`: Error message removed 45

1994/05/05 ltbibl.dtx v1.0c
- `\@citex`: Set switch for warning and end of run. 308
- `\nocite`: Do not write page number in `\nocite` warning message. 309
  - Set switch for warning and end of run. 309

1994/05/05 ltfinal.dtx v0.1g
- General: Added empty errhelp. 379
- `\errhelp`: Set error help empty. 385

1994/05/05 ltfntcmd.dtx v3.3c
- `\@@math@egroup`: Corrected `\@fontswitch` and added saved versions 197
- General: Corrected `\@fontswitch` 191

1994/05/05 ltmiscen.dtx v1.0i
- General: Removed braces from ifnextchar and ifstar arguments 202

1994/05/07 lttab.dtx v1.0c
- `\@maxtab`: Changed `\@firsttab` to `\chardef` 245
  - Changed `\@maxtab` to `\chardef` 245
- General: Removed definition of `\+` 241
  - Removed surplus braces from `\@ifnextchar` constructs 241

1994/05/08 ltfntcmd.dtx v3.3d
- General: Removed `\@undefinedfonterror` 191
- `\normalsize`: Removed `\@undefinedfonterror` 197

1994/05/09 ltfntcmd.dtx v3.3f
- General: Replaced all `\next` by `\@let@token` and undo change 3.3e, whatever that was. 191

1994/05/10 ltdefns.dtx v1.0n
- General: (ASAJ) Added `\DeclareProtectedCommand`. 22
  - Added `\DeclareProtectedCommand` 29
  - Added `\makeatletter` and `\makeatother` ASAJ. 33
  - Removed braces around `\@ifundefined` argument. ASAJ. 27

1994/05/10 lterror.dtx v1.0n
- `\@latexerr`: (ASAJ) Added extra blank lines to `\@latexerr`. 42

1994/05/10 ltmiscen.dtx v1.0j
- `\@sverb`: Slight change in error message text. 208

1994/05/11 ltboxes.dtx v1.0f
- `\@begin@tempboxa`: Use new `\color@setgroup` concept. 233
- `\@iiiminipage`: Use new `\color@setgroup` concept. 238
- `\@mpfootnotetext`: Use new `\color@setgroup` concept. 239
  - Use new `\normalcolor` and `\@finalstrut`. 239
- General: Superfluous braces removed from several commands 233
- `\color@setgroup`: macro added for colour support 234
- `\endminipage`: Use new `\color@setgroup` concept. 239

1994/05/11 ltclass.dtx v1.0c
- `\endfilecontents`: Add checks for form feed and tab 375

1994/05/11 ltdirchk.dtx v1.0e
- General: Add `\ProvidesFile` as used in fd files. 3

1994/05/11 lterror.dtx v1.0o
- `\@latexerr`: (ASAJ) Removed one of the extra blank lines to `\@latexerr`. 42

1994/05/11 ltlogos.dtx v1.0o
- `\LaTeX`: Use `\DeclareProtectedCommand`. ASAJ. 59
- `\LaTeXe`: Use `\DeclareProtectedCommand`. ASAJ. 59

1994/05/11 ltoutenc.dtx 1.5a
- General: Made T1 and OT1 generate packages rather than def files. Renamed the ‘package’ module to ‘teststy’. 72

1994/05/11 ltoutenc.dtx v1.5a
- General: Reimplemented `\DeclareTextCommand` using `\@changed@cmd` and `\DeclareProtectedCommand`. 72
  - Renamed the commands again. Made the encoding part of the command syntax. Added the `\DeclareTextCommand` interface. Used `\DeclareProtectedCommand`. 69
- `\DeclareTextAccent`: Reimplemented using `\DeclareTextCommand`. 74

1994/05/11 ltspace.dtx v1.0o
- `\,`: Use `\DeclareRobustCommand`. ASAJ. 57
- `\hspace`: Use `\DeclareRobustCommand`. ASAJ. 57

1994/05/12 ltboxes.dtx v1.0g
- `\@finalstrut`: macro added 240
- `\fbox`: New definition, merged with `\framebox` 236
- `\framebox`: Merged `\fbox` and `\framebox` 236
- `\normalcolor`: macro added for colour support 234

1994/05/12 ltdefns.dtx v1.0p
- General: (ASAJ) Fixed a bug with

\relax which was using \@gobble before defining it. 22
Fixed a bug with \relax which was using \@gobble before defining it. 29

1994/05/12 ltfssbas.dtx v2.1j
General: New baselinestretch concept 107
Replaced hand-protected commands by \DeclareRobustCommand defs 107
\f@linespread: New macro 114
\fontencoding: Use \DeclareRobustCommand. 112
\fontfamily: Use \DeclareRobustCommand. 113
\fontseries: Use \DeclareRobustCommand. 113
\fontshape: Use \DeclareRobustCommand. 113
\fontsize: Redefined to use \set@fontsize 114
\linespread: New macro 113
\mathversion: Use \DeclareRobustCommand. 114

1994/05/12 ltfssdcl.dtx v2.1g
General: Allow \relax as undefined command 149
Allow \relax'ed cmds to be declared 149

1994/05/12 ltfssini.dtx v2.1i
General: Moved \fontencoding to fam.dtx 168
Moved \fontfamily to fam.dtx 168
Moved \fontseries to fam.dtx 168
Moved \fontshape to fam.dtx 168
Moved \fontsize to fam.dtx 168
Moved \mathversion to fam.dtx 168
Moved \selectfont to tracefnt.dtx 168

1994/05/12 ltfsstrc.dtx v2.3f
\selectfont: Use \DeclareRobustCommand 128

1994/05/12 ltoutenc.dtx 1.5a
General: Removed the \SaveAtCatcode and \RestoreAtCatcode commands. 81
Rewrote for the new syntax. 81, 82

1994/05/12 ltoutput.dtx v1.0p
\@writesetup: \normalcoloradded 335
General: \normalcoloradded in various places (DPC). 314

1994/05/13 ltboxes.dtx v1.0h
\@arrayparboxrestore: New accent system, use \let not \def 238

1994/05/13 ltcounts.dtx v1.0f
General: Removed \@Ialph 105
Removed \@ialph 105

1994/05/13 ltdefns.dtx v1.0q
General: (ASAJ) Renamed \DeclareProtectedCommand to \DeclareRobustCommand. Removed \@if@short@command. 22
(ASAJ) Replaces \space by ' ' in \csname. 22
Renamed \DeclareProtectedCommand to \DeclareRobustCommand. Removed \@if@short@command. Moved to after the definition of \@gobble. 29

1994/05/13 ltdefns.dtx v1.0r
General: (ASAJ) Added logging message to \DeclareProtectedCommand. 22
Added logging message to \DeclareProtectedCommand. 29

1994/05/13 ltdefns.dtx v1.0s
General: (ASAJ) Added \@backslashchar. 22
(ASAJ) Coded \@ifdefinable more efficiently. 22
Coded more efficiently, thanks to FMi. 27

1994/05/13 ltfiles.dtx LaTeX2e
\listfiles: Stop \listfiles being run twice 67

1994/05/13 ltfiles.dtx v1.0g
\document: Added execution of \every@size 62

1994/05/13 ltfinal.dtx v0.1h
General: Added package ot1enc, and defined \@acci, \@accii and \@acciii. 379

1994/05/13 ltfinal.dtx v1.0h
General: Added output enc stuff 383

1994/05/13 ltfloat.dtx v1.0g
\@footnotetext: (DPC) Add new style colour support: \normalcolor 304
(DPC) Use \@finalstrut 304
\@xfloat: (DPC) Use \normalcolor 295

1994/05/13 ltfntcmd.dtx v3.3g
General: Replaced \@protecteddef by \DeclareRobustCommand 191

1994/05/13 ltfssbas.dtx v2.1k
General: Remove File identification 'typeout' 107

1994/05/13 ltfssbas.dtx v2.1l
\DeclareFontEncoding: Init encoding change command 110
\define@newfont: Use \@input@ for fd files 117

1994/05/13 ltfssdcl.dtx v2.1h
General: Removed file identification typeout 149

1994/05/13 ltfssini.dtx v2.1j
General: Removed file identification typeout 168

1994/05/13 ltfsstrc.dtx v2.3g
General: Removed typeouts as \ProvidesPackage writes to

log. 125

1994/05/13 ltoutenc.dtx v1.5b
- General: Added `\{`, `\}` and `\$`. . . 69
  - Renamed `\DeclareProtectedCommand` to `\DeclareRobustCommand`. 69
  - Replaces `\space` by ‘ ’ in `\csname`. 69

1994/05/13 ltpictur.dtx v0.1d
- General: Removed surplus braces from `\@if..` constructions . . 260

1994/05/13 lttab.dtx v1.0d
- `\@contfield`: Colour support . . . 246
- `\@startfield`: Colour support . . 246
- `\@stopfield`: Colour support . . . 246
- `\a`: moved to ltoutenc 245

1994/05/14 fontdef.dtx v2.1f
- General: Removed .def files. 175

1994/05/14 ltfssbas.dtx v2.1m
- `\enc@update`: Macro added 113

1994/05/14 ltfssbas.dtx v2.1n
- General: Set defaults for all `\f@`. . . 114
- `\DeclareErrorFont`: Don’t set `\f@encoding` 118
- `\DeclareFontEncoding`: Log if encoding is redeclared 110
  - Only init enc change cmd when new encoding 110

1994/05/14 ltfssini.dtx v2.1k
- General: Init error font just before checking for fontdef.cfg 171
- `\p@reset@font`: Remove surplus braces 170

1994/05/14 ltfsstrc.dtx v2.3h
- `\selectfont`: Added `\enc@update` 129

1994/05/14 ltoutenc.dtx 1.5d
- General: Moved the driver to the top. 72

1994/05/14 ltoutenc.dtx v1.5c
- General: Added the fontenc package 93
  - Added the fontenc package. . . 69
  - Fixed a bug which caused an infinite loop if `\f@encoding` was incorrectly set. 69, 72
  - Moved fontsmpl to its own dtx file. 69

1994/05/14 ltoutenc.dtx v1.5d
- General: Rewrote `\DeclareTextCommand` to define its argument to use the current encoding by default, rather than the encoding provided to `\DeclareTextCommand`. 69, 72
  - Tidied up the documentation. . 69

1994/05/14 ltoutenc.dtx v1.5e
- General: Replaced `\ENC@cmd` by `\ENC-cmd`. 69

1994/05/15 ltfssbas.dtx v2.1o
- General: encoding cmds changed to enc-cmd 107

1994/05/16 ltalloc.dtx v1.1a
- General: (ASAJ) Split from ltinit.dtx. 34

1994/05/16 ltcntrl.dtx v1.0a
- General: (ASAJ) Split from ltinit.dtx. 35

1994/05/16 ltdefns.dtx v1.1a
- General: (ASAJ) Split from ltinit.dtx. 22

1994/05/16 lterror.dtx v1.1a
- General: (ASAJ) Completely new error interface. 38
  - (ASAJ) Split from ltinit.dtx. . 38

1994/05/16 ltfinal.dtx v1.0i
- General: moved output enc stuff to lfonts 383

1994/05/16 ltfssbas.dtx v2.1p
- `\fontsize`: Pass `\baselinstretch` not `\f@linespread` 114
- `\linespread`: Remove surplus braces 113

1994/05/16 ltfssini.dtx v2.1m
- `\@acciii`: Define saved versions of accents 172

1994/05/16 ltlogos.dtx v1.1a
- General: (ASAJ) Split from ltinit.dtx. 59

1994/05/16 ltmath.dtx v1.0k
- `\ensuremath`: Use `\DeclareRobustCommand` and add extra braces in math mode 216

1994/05/16 ltoutenc.dtx 1.5h
- General: `\pounds` was still using u rather than ui shape. 81

1994/05/16 ltoutenc.dtx v1.5f
- General: enc files now have uc encoding name parts (FMi) 69
  - Revert code so that the encoding given is used in `\DeclareTextCommand` (FMi) 69

1994/05/16 ltoutenc.dtx v1.5g
- General: Made fontenc.sty use the new mixed-case encoding files. 69
  - Removed the lowercasing of the filename. 93

1994/05/16 ltoutenc.dtx v1.5h
- General: Added `\NG`, `\ng`, `\TH`, `\th`, `\DH`, `\dh`, `\DJ` and `\dj`. 69
  - Added `\r` (ring accent) and `\k` (ogonek) accents. 69
  - Fixed a bug with `\pounds`. . . . 69
  - Removed `\P` from the OT1 definitions file. 69

1994/05/16 ltoutenc.dtx v1.5i
- General: Fixed a bug with `\d`. . . 69

1994/05/16 ltoutput.dtx v1.0q
- `\@writesetup`: Changed setting of accents (FMi): with the new encoding setup they can use `\let`. It could also use the new internal commands? 336

- General: Changed setting of accents (FMi). 314
- 1994/05/16 ltpar.dtx v1.1a
  - General: (ASAJ) Split from ltinit.dtx. 46
- 1994/05/16 ltplain.dtx v1.0h
  - General: Comment out encoding specific commands 19
    - Remove `\@acci` and friends again 19
    - Remove unnecessary def for `\item` 19
  - `\loop`: Use Kabelschacht method 18
  - `\m@th`: Remove unnecssary space 18
- 1994/05/16 ltspace.dtx v1.1a
  - General: (ASAJ) Split from ltinit.dtx. 48
- 1994/05/17 ltclass.dtx v1.0e
  - `\@use@ption`: Execute option after removing from list, not before 370
- 1994/05/17 ltdefns.dtx 1.1b
  - General: (ASAJ) Added the `\@protect@...` commands. 30
- 1994/05/17 ltdefns.dtx v1.1b
  - General: (ASAJ) Added definitions for protect. 22
    - (ASAJ) Removed warnings and logging to lterror.dtx. 22
    - Added the discussion of protected commands, defined the values that `\protect` should have. 29
- 1994/05/17 ltdefns.dtx v1.1c
  - General: (ASAJ) Redid definitions for protect. 22
- 1994/05/17 lterror.dtx v1.1b
  - General: (ASAJ) Moved error stuff from ltdefns.dtx. 38
- 1994/05/17 ltfssini.dtx v2.1n
  - `\copyright`: Really add extra braces 170
  - `\nfss@text`: Added braces to allow use in subscripts 170
- 1994/05/17 ltmath.dtx v1.0i
  - General: Replaced `\let` by `\gdef`, for indirect definition. 214
- 1994/05/17 ltoutenc.dtx v1.5j
  - General: Added braces to `\pounds` so it works as a subscript. 69
- 1994/05/18 ltdefns.dtx 1.1c
  - General: (ASAJ) Renamed the commands, and removed one which is no longer needed. 30
- 1994/05/18 ltdefns.dtx v1.1c
  - General: Redid the discussion and definitions, in line with the proposed new setting of `\protect` in the output routine. 29
- 1994/05/18 ltfinal.dtx v0.1j
  - General: Corrected the lccode for d-bar. 379
- 1994/05/18 ltlogos.dtx v1.1b
  - General: (ASAJ) Added the TeX logo. 59
    - (ASAJ) Made the LaTeX 2$_\varepsilon$ logo use the text font ‘2’ rather than the math font ‘2’. 59
- 1994/05/18 ltoutenc.dtx v1.5k
  - General: Made dotted-i produce ‘i’. 69
    - Removed braces from `\pounds` and `\dollar`. 69
    - Replaced `\defaultencoding` with `\encodingdefault`. 69
- 1994/05/19 ltbibl.dtx v1.1a
  - General: Initial version of ltbibl.dtx, split from ltidxbib.dtx 307
- 1994/05/19 ltcounts.dtx v1.1a
  - General: Extracted file from ltcntlen. 103
- 1994/05/19 ltdefns.dtx v1.1d
  - General: (RmS) Added definitions for `\@namedef` and `\@nameuse` again. 22
- 1994/05/19 ltfinal.dtx v0.1k
  - General: Removed `\makeat...` 379
- 1994/05/19 ltidxglo.dtx v1.1a
  - General: Initial version of ltidxglo.dtx, split from ltidxbib.dtx 305
- 1994/05/19 ltlength.dtx v1.1a
  - General: Extract file ltlength from ltcntlen. 106
- 1994/05/19 ltpageno.dtx v1.1a
  - General: Extract file ltpageno from ltcntlen. 198
- 1994/05/19 ltplain.dtx v0.1k ltfinal
  - `\showoutput`: used `\maxdimen` not 99999 20
  - `\showoverfull`: used `\@ne` not 1 . 20
- 1994/05/19 ltxref.dtx v1.1a
  - General: Extract file ltxref from ltcntlen. 199
- 1994/05/1g fontdef.dtx v2.1g
  - General: Removed `\DeclareFontEncoding` for ot1 and t1 and input .def files instead 175
- 1994/05/2 ltdefns.dtx v1.1f
  - `\renewcommand`: Removed surplus `\space` in error 26
  - `\renewenvironment`: Removed surplus `\space` in error 27
- 1994/05/20 ltdefns.dtx v1.1e
  - General: Changed command name from `\@checkcommand` to `\CheckCommand`. 22
  - `\CheckCommand`: Changed name from `\@checkcommand` to `\CheckCommand`. 28
- 1994/05/20 lterror.dtx v1.1c
  - General: (ASAJ) Added `\@latex@info@no@line`. 38
    - (ASAJ) Added missing full stops. 38

(ASAJ) Fixed a bug with `\@inmatherr`. 38

1994/05/20 ltfinal.dtx v0.1l
General: Use new font warning commands 381

1994/05/20 ltfloat.dtx v1.0h
`\@endfloatbox`: Restore outer value of @nobreak switch. . . 297

1994/05/20 ltfntcmd.dtx v3.3h
General: Use new error commands 191

1994/05/20 ltfssbas.dtx v2.1q
General: Use new error commands 107

1994/05/20 ltfsstrc.dtx v2.3i
General: Use new error command names 125

1994/05/20 ltmiscen.dtx v1.0l
`\@writefile`: Added correct setting of `\protect`. 204

1994/05/20 ltmiscen.dtx v1.0m
General: Use new warning commands 202

1994/05/20 ltoutput.dtx v1.0s
`\@writesetup`: Added setting of `\protect` during `\shipout`. . 335
General: Added setting of `\protect` during `\shipout`. 314

1994/05/20 ltpage.dtx v1.0d
`\markright`: Changed setting for `\protect`. 312

1994/05/20 ltsect.dtx v1.0c
General: Correct setting of `\protect`. 289
`\addcontentsline`: Correct setting of `\protect`. 289

1994/05/21 ltbibl.dtx v1.1b
General: Use new warning commands 307

1994/05/21 lterror.dtx v1.1d
General: (ASAJ) Made the error commands robust. 38

1994/05/21 ltfiles.dtx v1.0h
General: Use new error commands 60

1994/05/21 ltlists.dtx v1.0f
General: Use new error commands 219

1994/05/21 ltmiscen.dtx v1.0n
General: Use new error commands 202

1994/05/21 ltsect.dtx v1.0d
General: Use new error commands 282

1994/05/21 lttab.dtx v1.0f
General: Use new error commands 241

1994/05/21 ltxref.dtx v1.1b
General: Use new warning commands 199
`\newlabel`: Use new warning commands 200

1994/05/22 ltclass.dtx v1.0f
General: Use new warning and error commands 362

1994/05/22 ltdefns.dtx v1.1f
General: Use new warning and error cmds 22

1994/05/22 lterror.dtx v1.1e
General: (ASAJ) Replaced bgroup by begingroup in error messages, to stop extra mathords creeping into math mode. . . . 38

1994/05/22 lterror.dtx v1.2a
General: (ASAJ) Made `\GenericError`, `\GenericWarning` and `\GenericInfo` robust. 38
(ASAJ) Replaced `\@generic@message` and `\@generic@error` by `\GenericError`, `\GenericWarning` and `\GenericInfo`. 38
(ASAJ) Replaced `\\` and tilde by `\MessageBreak` and `\space`. . . 38
(ASAJ) Replaces `\string` by `\protect` in some messages. . 38

1994/05/22 lterror.dtx v1.2d
`\GenericError`: (DPC) Alternative version added for old TeXs . . 38
(DPC) New version using long command name. 38

1994/05/22 ltfloat.dtx v1.0i
General: Use new warning commands 291

1994/05/22 ltoutput.dtx v1.0t
General: Changed warnings and infos to new commands. 314

1994/05/22 ltpictur.dtx v0.1e
General: Use new warning cmds . 260

1994/05/23 ltclass.dtx v1.0h
`\NeedsTeXFormat`: Don't stop completely when format is wrong 371
`\usepackage`: Remove argument if possible 371

1994/05/23 ltdirchk.dtx v1.0f
General: Document `\@TeXversion` 1

1994/05/23 ltfsstrc.dtx v2.3j
General: Removed def of `\f@warn@break` 139

1994/05/23 ltoutput.dtx v1.0u
`\@activechar@info`: Added `\MessageBreak` 335
`\@writesetup`: Changed resetting of `\protect` after shipout to use `\aftergroup` 335
General: Added `\MessageBreak`. . 314
Changed resetting of `\protect` after shipout. 314

1994/05/24 lterror.dtx v1.2e
`\@latex@info@no@line`: Macro added 41

1994/05/24 lterror.dtx v1.2f
General: (DPC) wrap long lines . 38

1994/05/24 ltfntcmd.dtx v3.3i
General: Tidying and typos fixed 191

1994/05/24 ltmiscen.dtx v1.0q
`\@currenvline`: Use `\@empty` as outer default 205

1994/05/25 ltdirchk.dtx v1.0g
`\filename@parse`: Mac parser had " typo for : 9

- 1994/05/25 ltfntcmd.dtx v3.3j
  - General: Insertion of `\aftergroups` to implement `\nocorr` moved to the end of the group 191
  - `\check@icr`: Macros added 194
  - `\check@nocorr@`: Insertion of `\aftergroups` moved and defaults set up for efficiency 194
  - `\DeclareTextFontCommand`: `\expandafter` inserted 193
    - Insertion of `\aftergroups` moved 193
- 1994/05/25 ltoutput.dtx v1.0v
  - General: Extra documentation. 314
- 1994/05/25 ltsect.dtx v1.0e
  - `\@dottedtocline`: Put braces around argument 4 (the actual toc entry) to avoid font (and possibly other) changes leaking out to the leaders. 290
- 1994/05/25 ltthm.dtx v1.0c
  - General: Modify documentation 279
- 1994/05/25 ltvers.dtx v1.0d
  - General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here) 21
- 1994/05/25 ltxref.dtx v1.1c
  - General: Modify documentation 199
- 1994/05/26 ltfiles.dtx LaTeX2e
  - `\@missingfileerror`: Modify message format 66
- 1994/05/26 ltlogos.dtx v1.1c
  - General: Remove `\SLiTeX` logo 59
- 1994/05/26 ltplain.dtx v1.1m
  - `\iterate`: (CAR) added `\long` 18
  - `\underbar`: (CAR/FMi) changed to use box `\tw@` 19
- 1994/05/26 ltplain.dtx v1.1p
  - `\underbar`: (DPC) changed to use `\sbox` 19
- 1994/05/26/16 ltmiscen.dtx v1.0r
  - General: `\literal` removed 209
- 1994/05/29 ltfssdcl.dtx v2.1j
  - General: Use new error commands 149
- 1994/05/31 ltfinal.dtx v1.0n
  - General: Renamed lthyphen.* to lthyphen.*. 379
- 1994/06/01 ltboxes.dtx v1.0i
  - `\@frameb@x`: Macro added. 236
  - `\@iframebox`: New version, so `\width` is correct in `\framebox` 236
  - `\fbox`: New version, using `\@frameb@x` 236
  - `\framebox`: New version, so `\width` is correct in `\framebox` 236
- 1994/06/01 ltlogos.dtx v1.1d
  - `\LaTeX`: Add `\m@th` to force math size calculations 59
- 1994/06/01 ltoutput.dtx v1.0w
  - General: Tidied up typesetting. 314
- 1994/06/08 ltfinal.dtx v1.0m
  - General: Add patch file system 383
- 1994/06/09 ltfinal.dtx v1.0n
  - General: For TEX2, do not set codes for higher half of character table. 380, 382
- 1994/06/09 ltfntcmd.dtx v3.3k
  - General: Tidying and typos fixed in documentation 191
- 1994/06/18 ltfntcmd.dtx v3.3l
  - General: Added check for empty text 191
  - `\check@nocorr@`: Added check for empty text 194
- 1994/06/22 ltfntcmd.dtx v3.3m
  - General: Removed space from `\nfss@text` 191
    - Renamed `\check@nocorr` 191
  - `\check@nocorr@`: Renamed `\check@nocorr` to `\text@command` to improve `\long` error message 194
  - `\DeclareTextFontCommand`: Removed space from `\nfss@text` 193
- 1994/06/22 ltmath.dtx v1.2t classes
  - `\mathindent`: Set `\mathindent` at the end of the class instead of at begin document 217
- 1994/07/20 ltlogos.dtx v1.1e
  - `\LaTeX`: Save a few tokens 59
  - `\LaTeXe`: Save a few tokens 59
- 1994/07/20 ltpage.dtx v1.0h
  - `\sloppy`: Save a few tokens 313
- 1994/09/16 ltfssbas.dtx v2.1s
  - `\nfss@catcodes`: Reset [and] as well, just in case 118
- 1994/10/07 ltoutenc.dtx v1.5l
  - General: Moved the ogonek accent. 69
- 1994/10/11 ltdirchk.dtx v1.0h
  - `\@TeXversion`: Check for TeX3.14 10
  - General: Modify all of ltxcheck again 10
- 1994/10/12 ltsect.dtx v1.0f
  - General: Doc. typos 282
- 1994/10/14 fontdef.dtx v2.2a
  - General: New coding 173
- 1994/10/14 ltfssini.dtx v2.2a
  - General: New coding for cfg files 168
- 1994/10/14 ltmiscen.dtx v1.0s
  - General: Move math to other file 202
- 1994/10/14 ltplain.dtx v1.1a
  - General: Moved code to other files. 11
- 1994/10/15 ltfssbas.dtx v2.1t
  - `\extract@alph@from@version`: Warn if math alpha is used outside math 122
- 1994/10/18 ltboxes.dtx v1.0j
  - `\@frameb@x`: `\leavevmode` added 236
  - `\@iframebox`: `\leavevmode` moved to `\@frameb@x` 236
  - `\@parboxto`: Macro added to remove misuse of `\@empty` 237
  - General: stuff from ltpatch done 233
  - `\fbox`: `\long` added 236
  - `\mbox`: `\long` added 233

- `\sbox`: `\long` added 235

1994/10/18 ltclass.dtx v1.0j
- General: Move `\listfiles` to ltfiles.dtx 377

1994/10/18 ltdefns.dtx v1.2a
- `\@star@or@long`: macro added 24
- General: Add extra test for `\endgraf` 22
  - Add star-forms for all commands 22
- `\renew@environment`: reset end command 27

1994/10/18 ltfiles.dtx v1.0i
- `\listfiles`: code moved here from ltclass 67

1994/10/18 ltoutenc.dtx v1.5l
- General: Added new definitions of `\patterns` and `\hyphenation`. 78

1994/10/18 ltoutenc.dtx v1.5m
- General: Added new definitions of `\patterns` and `\hyphenation`. 69

1994/10/18 ltsect.dtx v1.0g
- `\@dottedtocline`: Added `\normalcolor` for page number 290
- General: Added `\normalcolor` 282

1994/10/19 ltfssbas.dtx v2.1t
- `\DeclareFontEncoding`: Add missing `\relax`. 109

1994/10/23 ltfsstrc.dtx v23.k
- `\every@math@size`: Renamed to `\every@math@size` 131

1994/10/23 ltmath.dtx v1.0l
- `\@eqnnum`: Added `\normalcolor` since `\eqno` introduces a subgroup of the displayed math group 214
- `\ensuremath`: Remove extra braces: but see p 168 of Leslie's book 216

1994/10/24 ltboxes.dtx v1.0k
- `\fbox`: Inner braces added (to fix latex/1061) 236

1994/10/25 fontdef.dtx v2.2c
- General: Added OMSenc.def 175

1994/10/25 ltboxes.dtx v1.0l
- `\@isavepicbox`: missing percent (moved from ltpatch) 235

1994/10/25 ltdefns.dtx v1.2b
- General: Documentation improvements 22

1994/10/25 ltoutenc.dtx 1.6a
- General: Added `\textdollar`, `\textlbrace`, `\textrbrace`, `\textsterling`, `\textunderline`. 82
  - Removed `\textlbrace`, `\textrbrace`, `\textunderline` to give them their proper names. 82

1994/10/25 ltoutenc.dtx v1.6a
- General: Added `\ProvideTextCommand`, `\UseTextSymbol`, `\UseTextAccent`, `\DeclareTextSymbolDefault`, `\DeclareTextAccentDefault`, `\DeclareTextCommandDefault`, and `\ProvideTextCommandDefault`. 69
  - Added the `\Provide` commands, and the default definitions. 72
  - Added the defaults. 79
  - Added the files OT1enc.def, T1enc.def and OMSenc.def. 78
  - Added the OMS encoding. 86

1994/10/27 ltoutenc.dtx 1.6b
- General: Added `\textasciicircum` `\textasciitilde` `\textbackslash` `\textbar` `\textbraceleft` `\textbraceright` `\textcompwordmark` `\textemdash` `\textendash` `\textexclamdown` `\textgreater` `\texthyphenchar` `\texthyphen` `\textless` `\textquestiondown` `\textquotedblleft` `\textquotedblright` `\textquotedbl` `\textquoteleft` `\textquoteright` `\textunderscore` `\textvisiblespace` 82
  - Added: `\textemdash` `\textendash` `\textexclamdown` `\texthyphenchar` `\texthyphen` `\textquestiondown` `\textquotedblleft` `\textquotedblright` `\textquoteleft` `\textquoteright` 81

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`. 72

1994/10/27 ltoutenc.dtx v1.6b
- General: Added `\textbackslash`. 86
  - Added more defaults for OT1. 79
  - Removed the enc.def files 69
  - Removed the files OT1enc.def, T1enc.def and OMSenc.def. 78
  - Renamed `\textlbrace` to `\textbraceleft` and `\textrbrace` to `\textbraceright`. 86

1994/10/29 ltmath.dtx 1.0m
- General: ASAJ: Added `\DeclareMathOperator`. 210
  - ASAJ: Tidied up documentation. 214

1994/10/29 ltmath.dtx v1.0m
- General: ASAJ: Added `\mathellipsis`, `\mathdollar` and `\mathsterling`. 214
  - ASAJ: Removed `\dag`, `\ddag`. 214
  - ASAJ: Renamed `\S` and `\P` to `\mathsection` and `\mathparagraph` and made them `\mathchardefs`. 214

1994/10/29 ltoutenc.dtx v1.6c
- General: Added commands like `\dots` for use in text and math. 79
  - Renamed `\P`, `\S`, `\dag` and `\ddag` to `\textparagraph`, `\textsection`, `\textdagger` and `\textdaggerdbl`. 69

1994/10/30 ltdefns.dtx v1.2c
- `\@onelevel@sanitize`: Macro added 33
- General: (CAR)`\@onelevel@sanitize` added 22

1994/10/30 ltdefns.dtx v1.2f
- General: (DPC)`\newwrite`'s moved to ltfiles 22

1994/10/30 ltmath.dtx v1.0n
- General: ASAJ: Moved the new commands to ltoutenc. 214

1994/10/30 ltoutenc.dtx v1.6d
- General: Added `\DeclareTextCompositeCommand`. 69
  - Added `\textcircled`. 69, 80, 86
  - Added `\t`. 80
  - Added math commands. 69
  - Added OML encoding. 69, 80
  - Added the OML encoding. 87
  - Made `\textless` and `\textgreater` come from OML. 80
  - Moved math commands here from ltmath. 81
  - Removed `\textregistered`. 80
  - Rewrote `\copyright` to use `\textcircled`. 80

1994/10/31 fontdef.dtx v2.2d
- General: Added OMLenc.def ... 175

1994/10/31 fontdef.dtx v2.2e
- General: ... and moved further down 175

1994/10/31 ltfloat.dtx v1.1a
- `\@dblfloat`: Major changes since two-column and one-column cases merged 294
- `\@dblflset`: Macro added 294
  - Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved 294
- `\@endfloatbox`: (DPC/CAR) Extra box added to remove colour resetting from vmode 297
- `\@floatboxreset`: Macro added 295
- `\@footnotetext`: (DPC/CAR) Move colour setting to output routine 304
- `\@savemarbox`: (DPC/CAR) Extra box added for colour 299
- `\@setfps`: Macro added 294
- `\@xdblfloat`: Macros removed: `\@dbflt`, `\@xdblfloat` 297
- `\@xfloat`: (DPC/CAR) Extra box added to remove colour resetting from vmode 295
  - Major changes, removing setting of local variables, space hacks etc; two-column and one-column cases merged 294
  - Reset hook added 295
- `\@xympar`: (DPC/CAR) Extra box added since needed for floats 300
- `\fps@dbl`: Macro added 294

1994/10/31 ltoutput.dtx v1.1a
- `\@makecol`: (DPC/CAR) Colour resetting moved to here 332
- `\@topnewpage`: (DPC/CAR) Extra box added to remove colour resetting from vmode 326
  - (DPC/CAR) Use `\color@begingroup` for colour 326
  - (DPC/CAR) Use `\normalcolor` 326

1994/11/02 ltoutenc.dtx v1.6d
- General: Wrapped lines longer than 70 characters. 69

1994/11/03 ltclass.dtx v1.0k
- General: Move `\@missingfileerror` to ltfiles 365

1994/11/03 ltdirchk.dtx v1.0i
- General: Generate an error if latex.ltx not used with clean initex 1

1994/11/03 ltfiles.dtx v1.0j
- `\@missingfileerror`: Move here from ltclass 66

1994/11/04 ltboxes.dtx v1.0m
- `\@mpfootnotetext`: Added `\protected@edef`. ASAJ. 239

1994/11/04 ltdefns.dtx v1.2e
- General: Added `\set@display@protect` to `\typeout`. ASAJ. 22
  - Added commands for setting and restoring `\protect`. ASAJ. 31
  - Rewrote protected short commands using `\x@protect`. ASAJ. 30

1994/11/04 lterror.dtx v1.2g
- General: Added `\set@display@protect` to `\Generic*` commands. ASAJ. 38

1994/11/04 ltfiles.dtx v1.0k
- `\nofiles`: Added setting of `\protected@write`, `\makeindex` and `\makeglossary` to `\nofiles`. ASAJ. 63
- `\protected@write`: Macro added ASAJ. 64

1994/11/04 ltfloat.dtx v1.1b
- `\@footnotetext`: (ASAJ) Added `\protected@edef`. 304
- `\footnotemark`: Added `\protected@xdef` to `\footnotemark`. 304

- 1994/11/04 ltidxglo.dtx v1.1b
  - `\@wrglossary`: Added `\protected@write` to `\@wrglossary`. 306
  - `\@wrindex`: Added `\protected@write` to `\@wrindex`. 306
  - General: Removed `\if@filesw` from `\makeindex`. 305
  - `\makeglossary`: Removed `\if@filesw` from `\makeglossary`. 306
- 1994/11/04 ltmiscen.dtx v1.0t
  - `\@writefile`: Removed setting of `\protect`. ASAJ. 204
- 1994/11/04 ltoutenc.dtx v1.6f
  - General: Added `\_`. 80
    - Added `\mathunderscore`. 81
- 1994/11/04 ltpage.dtx v1.0e
  - `\markright`: Added `\@unexpandable@protect`. ASAJ. 312
- 1994/11/04 ltsect.dtx 1.0h
  - `\@sect`: (ASAJ) Added `\protected@edef`. 285
  - General: (ASAJ) Added `\protected@xdef` to `\thanks`. 282
- 1994/11/04 ltsect.dtx v1.0h
  - General: Added `\protected@write` to `\addtocontents`. ASAJ. 289
  - `\addcontentsline`: Added `\protected@write` to `\addcontentsline`. ASAJ. 289
- 1994/11/04 lttab.dtx v1.0h
  - `\@mkpream`: (ASAJ) Added `\@unexpandable@protect` to `\@mkpream`. 256
  - `\multicolumn`: (ASAJ) added `\set@typeset@protect`. 253
- 1994/11/04 ltxref.dtx v1.1d
  - `\label`: (ASAJ)Added `\protected@write` 200
  - `\refstepcounter`: (ASAJ)Added `\protected@edef` 200
- 1994/11/05 ltboxes.dtx v1.0n
  - `\@mpfootnotetext`: Colour resetting for footnotes moved to endminipage: as for main page. 239
  - `\color@endbox`: macro added for colour support 234
  - `\color@hbox`: macro added for colour support 234
  - `\endminipage`: Colour resetting for footnotes moved to here: as for main page. 239
- 1994/11/05 ltboxes.dtx v1.0o
  - `\@mpfootnotetext`: Colour groups restored here. 239
- 1994/11/05 ltfloat.dtx v1.1c
  - `\@dblflset`: Add compatibility with old version of `\@xfloat`. 294
  - `\@endfloatbox`: Use new `\color@hbox` concept. 297
  - `\@footnotetext`: Removed `\normalcolor` (again) 304
  - `\@savemarbox`: Use new `\color@hbox` concept. 299
  - `\@setfps`: Add compatibility with old version of `\@xfloat`. 294
  - `\@xfloat`: Add compatibility with old version of `\@xfloat`: but the arguments, provided at exorbitant cost, are now completely ignored 294
    - Use new `\color@hbox` concept. 295
  - `\@xympar`: Use new `\color@hbox` concept. 300
- 1994/11/05 ltoutenc.dtx v1.6g
  - General: Added setting of `\@typeset@protect` to `\patterns` and `\hyphenation`. 78
- 1994/11/05 ltoutput.dtx v1.1b
  - `\@topnewpage`: Use new `\color@hbox` concept. 326
  - `\@writesetup`: Change protect settings for new-style, protect-free aux-files. 335
    - Use new `\color@hbox` concept. 335
- 1994/11/05 ltoutput.dtx v1.1c
  - `\@begindvi`: Added macro 337
  - `\@begindvibox`: Added macro 324
  - `\@writesetup`: Add new `\AtBeginDvi` concept 335
  - `\AtBeginDvi`: Added macro 324
- 1994/11/06 ltfssbas.dtx v2.1u
  - `\cf@encoding`: New macro 114
  - `\DeclareFixedFont`: Renamed `\every@size` to `\every@math@size`. 108
- 1994/11/06 ltfssini.dtx v2.2b
  - `\@setsize`: Use `\@typeset@protect` 169
- 1994/11/06 ltfsstrc.dtx v2.3k
  - `\glb@currsize`: New implementation 130
  - `\try@simples`: New implementation 140
  - `\try@size@substitution`: New implementation 140
  - `\tryis@simple`: New implementation 140
- 1994/11/07 fontdef.dtx v2.2f
  - General: (DPC) Add `\DeclareMathSizes` declarations 177
    - (DPC) Updated to use `\ProvidesFile` 175
- 1994/11/07 ltfiles.dtx v1.0l
  - `\@unused`: move here from ltdefns, remove duplicate `\@mainaux` 61
- 1994/11/07 ltfiles.dtx v1.0m
  - `\document`: Renamed `\every@size` to `\every@math@size`. 62
- 1994/11/07 preload.dtx v2.1e
  - General: (DPC) Updated to use `\ProvidesFile` 188

1994/11/09 ltboxes.dtx v1.0p
- `\@finalstrut`: Revert `\finalstrut` to 2.09 equivalent (from ltpatch) 240
- General: more colour changes. . . 233

1994/11/09 ltfssbas.dtx v2.1v
- `\@vpt`: (DPC) macros added, from setsize.dtx 123
  - (DPC) reduce save stack usage latex/1742 123

1994/11/10 ltbibl.dtx v1.1c
- General: Fix `\nocite{*}` 307
- `\nocite`: Fix `\nocite{*}` 309

1994/11/10 ltmath.dtx v1.2v classes
- `eqnarray`: Added value of `\parskip` to `\abovedisplayskip` to compensate for negative `\topsep` 218

1994/11/10 ltoutput.dtx v1.1e
- `\@writesetup`: Modify `\protect` setting 335

1994/11/10 ltplain.dtx v1.1b
- General: (CAR) added patch to `\loop`. 11
- `\iterate`: (CAR) added extra `\relax` 18

1994/11/11 ltspace.dtx v1.2a
- `\\`: (DPC) Make robust 51

1994/11/12 ltfntcmd.dtx v3.3o
- `\normalsize`: Added `\MessageBreak` 197

1994/11/12 ltlists.dtx v1.2b ltspace
- `\endtrivlist`: Changed order of tests to make `\@noitemerror` correct: end of an era. 227

1994/11/12 ltmiscen.dtx v1.0u
- `center`: Changed end macro to `\def`: safer and consistent 206
- `flushleft`: Changed end macro to `\def`: safer and consistent 206
- `flushright`: Changed end macro to `\def`: safer and consistent 206

1994/11/12 ltplain.dtx v1.1c
- General: Comment out more encoding specific commands 19

1994/11/12 ltspace.dtx v1.2b
- `\addpenalty`: Corrected error message 55
- `\addvspace`: Corrected error message 55

1994/11/13 ltspace.dtx v1.2c
- `\addpenalty`: Recorrected error message 55
- `\addvspace`: Recorrected error message 55

1994/11/14 ltoutput.dtx v1.1f
- `\@begindvi`: Use normal box register: why a box? 337
- `\@begindvibox`: Use normal box register: why a box? 324
- `\@writesetup`: Modify new `\AtBeginDvi` concept 335
- General: Removed old definition of `\@testfp`. 314

1994/11/14 ltspace.dtx v1.2d
- `\\`: (DPC) Macro modified 51

1994/11/14 lttab.dtx v1.0i
- `\tabularnewline`: (DPC) Macro added 252

1994/11/16 fontdef.dtx v2.2h
- General: (DPC) Removed `\{` and `\}` 175

1994/11/17 ltboxes.dtx v1.0q
- General: `\@tempa` to `\reserved@a` 233

1994/11/17 ltclass.dtx v1.0l
- General: `\@tempa` to `\reserved@a` 362

1994/11/17 ltcntrl.dtx v1.0b
- General: `\@tempa` to `\reserved@a` 35

1994/11/17 ltdefns.dtx v1.0g
- General: `\@tempa` to `\reserved@a` 22

1994/11/17 ltdirchk.dtx v1.0j
- General: `\@tempa` to `\reserved@a` . 1

1994/11/17 lterror.dtx v1.2h
- General: `\@tempa` to `\reserved@a` 38

1994/11/17 ltfiles.dtx v1.0n
- General: `\@tempa` to `\reserved@a` 60

1994/11/17 ltfinal.dtx v1.0o
- General: `\@tempa` to `\reserved@a` 379

1994/11/17 ltfloat.dtx v1.1e
- General: `\@tempa` to `\reserved@a` 291

1994/11/17 ltfntcmd.dtx v3.3p
- General: `\@tempa` to `\reserved@a` 191

1994/11/17 ltfssbas.dtx v2.1w
- General: `\@tempa` to `\reserved@a` 107

1994/11/17 ltfssdcl.dtx v2.1m
- General: `\@tempa` to `\reserved@a` 149

1994/11/17 ltfsstrc.dtx v2.3l
- General: `\@tempa` to `\reserved@a` 125

1994/11/17 ltmath.dtx v1.0o
- General: `\@tempa` to `\reserved@a` 210

1994/11/17 ltmiscen.dtx v1.0v
- General: `\@tempa` to `\reserved@a` 202

1994/11/17 ltoutenc.dtx v1.6h
- General: (DPC) `\@tempa` to `\reserved@a` 69

1994/11/17 ltoutput.dtx v1.1h
- General: `\@tempa` to `\reserved@a`. 314

1994/11/17 ltpictur.dtx v1.0f
- General: `\@tempa` to `\reserved@a` 260

1994/11/17 ltsect.dtx v1.0i
- General: `\@tempa` to `\reserved@a` 282

1994/11/17 lttab.dtx v1.0j
- General: `\@tempa` to `\reserved@a` 241

1994/11/18 ltboxes.dtx v1.0r
- `\color@vbox`: macro added for colour support 234

1994/11/18 ltfinal.dtx v1.0n
- General: re-allow slots 127–255 382

1994/11/18 ltfssbas.dtx v2.1x
- General: (DPC) use `\reserved@f` not `\next` 107

1994/11/18 ltfssdcl.dtx v2.1m
- `\DeclareMathDelimiter`: (DPC) `\expandafter` instead of `\next` 162

- 1994/11/18 ltfsstrc.dtx v2.3m
  - General: \next to \reserved@f . 125
- 1994/11/18 ltmath.dtx v1.0p
  - \phantom: (DPC) colour support 211
    - (DPC) use \expandafter instead of \next 211
  - \prime@s: (DPC) use \@let@token instead of \next and \expandafter instead of \nxt 213
  - \smash: (DPC) colour support 212
    - (DPC) use \expandafter instead of \next 212
- 1994/11/21 ltfloat.dtx v1.1f
  - \@endfloatbox: Added reset of minipage flag 297
    - Corrected position of \outer@nobreak 297
  - \@marginparreset: Macro added 299
  - \@savemarbox: Added \@setminipage etc 299
    - Added resetting of size and font 299
    - Changed to \color@vbox 299
    - Use \@setnobreak etc 299
  - \@setminipage: Macro added 296
  - \@setnobreak: Macro added 296
  - \@xfloat: Added \@setminipage 295
    - Added resetting of size and font 295
    - Changed to \color@vbox so that large floats overflow at the bottom 295
    - Missing percents reinserted after 4, 8: these are not numbers. 294
    - Use \@setnobreak 295
  - \@xympar: Changed to \color@vbox 300
- 1994/11/21 ltoutput.dtx v1.1i
  - \@addtocurcol: Added \if@nobreak test before float box 344
  - \@specialoutput: Added \if@nobreak test 330
  - \@topnewpage: Changed to \color@vbox 326
- 1994/11/22 ltfssdcl.dtx v2.1o
  - General: wrap long lines 149
- 1994/11/22 ltoutenc.dtx v1.6i
  - General: Corrected \dots so that there's no kerning in monowidth fonts. 69
    - Corrected typo with \mathunderscore. 69
    - Fixed empty accents. Again. 69
- 1994/11/24 ltdefns.dtx v1.2h
  - \@newenv: Added test for \endgraf 27
- 1994/11/25 ltplain.dtx v1.1f
  - General: (DPC) Comment out lots of obsolete code 11
- 1994/11/26 ltfloat.dtx v1.1b
  - \footnote: (ASAJ) Added \protected@xdef. 303
- 1994/11/28 ltcntrl.dtx v1.0c
  - General: Documentation improvements 35
- 1994/11/30 ltfiles.dtx v1.0o
  - \@dofilelist: Macro added 68
  - \listfiles: Use \@dofilelist 67
  - \nofiles: There is no \@gobblethree... 63
- 1994/11/30 ltfssbas.dtx v2.1y
  - \fontshape: Use \@current@cmd in \@@enc@update. ASAJ. 113
- 1994/11/30 ltmath.dtx 1.0q
  - General: ASAJ: \DeclareMathOperator moved to AMSLaTeX. 210
- 1994/11/30 ltmiscen.dtx v1.0w
  - \enddocument: (DPC) Do warnings even for \nofiles 203
    - (DPC) Use \@dofilelist 203
- 1994/11/30 ltoutenc.dtx 1.7a
  - General: Redefined \a for the new scheme. 78
- 1994/11/30 ltoutenc.dtx v1.6g
  - General: Removed new definitions of \patterns and \hyphenation, since encoding-specific commands now expand in the mouth. 78
- 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. 69
    - Always load the enc.def file, so that the default encoding for the commands will change. 93
    - Redefined \@changed@cmd to expand in the mouth. 72
    - Removed \@changed@x@mouth since \@changed@x now expands in the mouth. 72
    - Rewrote \@text@composite so it allows an empty argument, or an argument containing lots of commands. 74
- 1994/12/01 ltfinal.dtx v1.0p
  - General: Renamed lthyphen.* to hyphen.*. 379
- 1994/12/01 lthyphen.dtx v1.0g
  - General: Rename lthyphen.ltx/cfg to hyphen.ltx/cfg 378
- 1994/12/01 ltplain.dtx v1.1g
  - General: (DPC) More doc changes 11
- 1994/12/02 fontdef.dtx v2.2i
  - General: Commented out \ldots. ASAJ. 173
- 1994/12/02 ltfssini.dtx v2.2c
  - \copyright: \copyright is now in ltoutenc. ASAJ 170
- 1994/12/02 ltlists.dtx v1.0e
  - \@trivlist: RmS: Added check for looping 226
- 1994/12/02 ltoutenc.dtx 1.7b
  - General: Redefined \a properly. 78

1994/12/02 ltoutenc.dtx v1.7b
  General: Fixed a bug with `\a`. . . 69
1994/12/04 lthyphen.dtx v1.0h
  General: Documentation edits for /1989 . . . 378
1994/12/05 ltoutenc.dtx v1.7c
  General: Added braces to `\textcircled`. . . . 69
1994/12/06 ltfssbas.dtx v2.1z
  `\DeclareFontEncoding`: use `\nfss@catcodes` . . . 109
  `\nfss@catcodes`: Added tab char as well . . . 118
1994/12/08 ltoutenc.dtx v1.7d
  General: Added `\null` and `\sh@ft` to `\b` and `\d`. . . . 69
1994/12/08 lttab.dtx v1.0k
  `\@array`: Add `\tabularnewline` . 251
  `\tabularnewline`: (DPC) Made it `\relax` . . . 252
1994/12/09 ltbibl.dtx v1.1d
  `\bibliographystyle`: (DPC) Allow use in preamble. . . . 309
1994/12/10 ltfloat.dtx v1.1g
  `\@dblfloat`: Old version reinstated temporarily . . . 294
  `\@dblflset`: Macro removed temporarily . . . 294
    Old version reinstated temporarily . . . 294
  `\@setfps`: Macro removed temporarily . . . 294
  `\@xdblfloat`: Macros reinserted temporarily . . . 297
  `\@xfloat`: Old version reinstated temporarily . . . 294
    Sanitisation added temporarily 294
  General: Some temps reinserted temporarily . . . 291
  `\fps@dbl`: Macro removed temporarily . . . 294
1994/12/10 ltfntcmd.dtx v3.3q
  `\@@math@egroup`: Don't read arguments . . . 197
  `\check@nocorr@`: Use `\space` command for comparison . . . 194
1994/12/10 ltfssdcl.dtx v2.1p
  `\document@select@group`: Surround with braces (add fourth arg) . . . 152
  `\select@group`: Surround with braces (add fourth arg) . . . 151
1994/12/10 ltoutenc.dtx v1.7e
  General: Added documentation for the OML encoding. . . . 69
    Replaced width with `\@width` and ditto height in vrules. . . . 69
1994/12/14 ltoutenc.dtx v1.7f
  General: Added braces to `\copyright` so it works unbraced in subscripts. . . . 69
    Added check for math mode in `\@changed@cmd`. . . . 69
    Commented out `\textasciicircum`, `\textasciitilde`, `\textbackslash`, `\textbar`, `\textgreater`, `\texthyphenchar`, `\texthyphen` and `\textless` to save memory. . . . 69
1995/01/12 ltmath.dtx v1.2y classes
  `\@eqnnum`: Added `\normalcolor` . 216
1995/03/03 ltoutenc.dtx 1.7g
  General: Corrected an error in documentation referring to the tabular rather than the tabbing environment. . . . 78
1995/04/02 ltfntcmd.dtx v3.3r
  `\@@math@egroup`: Read them again to be able to add `\relax`. . . 197
1995/04/02 ltfssdcl.dtx v2.1q
  `\document@select@group`: fix problem for pr/1275 . . . 152
  `\select@group`: fix problem for pr/1275 . . . 151
  `\set@mathdelimiter`: fix pr/1329 164
1995/04/02 ltfssini.dtx v2.2d
  `\not@math@alphabet`: add `\noexpand` to second part of message 169
1995/04/21 ltclass.dtx v1.0m
  `\DeclareOption*`: Made long /1498 . . . 368
  `\endfilecontents`: Close input check stream: latex/1487 . . . 375
1995/04/21 ltfinal.dtx v1.0q
  General: Allow initial patch level 0 384
1995/04/21 ltoutenc.dtx v1.7h
  General: Added `\null` `\k` latex/1274 . . . 69
1995/04/22 ltfiles.dtx v1.0p
  `\includeonly`: Allow blanks in argument . . . 64
1995/04/22 ltmiscen.dtx v1.0x
  General: Removed extra def of `\@gobble` . . . 202
1995/04/23 ltsect.dtx v1.0j
  `\addcontentsline`: Use `\contentsline` internally. . . . 289
1995/04/24 ltbibl.dtx v1.1e
  `\@citex`: Add `\mbox` to undefined case: latex/1239. . . . 308
1995/04/24 ltbibl.dtx v1.1f
  `\bibcite`: Make `\@onlypreamble` /1388. . . . 308
1995/04/24 ltcntrl.dtx v1.0d
  `\@for`: Dont expand second argument with `\edef`: /1317 (DPC) 37
1995/04/24 ltoutput.dtx v1.1j
  `\@tracemessage`: Do not add to kernel unless 'trace' specified 351
1995/04/24 ltoutput.dtx v1.1l
  `\@begindvibox`: Add `\vbox` latex/1392 . . . 324

`\@writesetup`: Reset \\ latex/1451 (DPC) . . . 336
1995/04/24 ltpage.dtx v1.0f
`\fussy`: reset `\emergencystretch` latex/1344 . . . 313
1995/04/24 ltplain.dtx v1.1h
`\newtoks`: Remove remaining `\outer` declarations. . . . 13
1995/04/24 ltxref.dtx v1.1e
`\newlabel`: Make `\@onlypreamble` for /1388. . . . 200
1995/04/25 ltdefns.dtx v1.2i
`\@check@c`: Make `\long` for latex/1346 . . . 29
`\new@environment`: Parse arguments slowly but safely /1507 27
1995/04/25 ltfiles.dtx v1.0q
`\document`: Removed execution of `\every@size` latex/1407 . . . 62
1995/04/25 ltsect.dtx v1.0k
`\@dottedtocline`: Added `\hbox` around dots. . . . 290
1995/04/27 ltboxes.dtx v1.0s
`\@frameb@x`: Move `\leavevmode` for graphics/1512 . . . 236
`\@iframebox`: Move `\leavevmode` for graphics/1512 . . . 236
`\@iirsbox`: Move `\leavevmode` for graphics/1512 . . . 240
`\@irsbox`: Move `\leavevmode` for graphics/1512 . . . 240
`\fbox`: Move `\leavevmode` for graphics/1512 . . . 236
`\raisebox`: Move `\leavevmode` for graphics/1512 . . . 240
1995/04/27 ltfiles.dtx v1.0r
`\document`: Added `\global` to support groups in hook . . . 63
1995/04/27 ltmiscen.dtx v1.0y
`\enddocument`: `\@checkend` moved after hook . . . 203
1995/04/27 ltplain.dtx v1.1i
General: Move `\hang` and `\textindent` to latex209.def . . . 19
1995/04/29 ltcntrl.dtx v1.0e
General: Moved init of `\protect` to ltdefns.dtx . . . 37
Removed unused defs for `\@setprotect` and `\@resetprotect` 37
1995/04/29 ltdefns.dtx v1.2j
`\protect`: Init `\protect` here . . . 31
1995/04/29 ltpar.dtx v1.1b
General: (TO) Comments cleanup. . . . 46
1995/05/02 ltsect.dtx v1.0l
`\@dottedtocline`: Don't reset to `\rmfamily` . . . 290
1995/05/03 ltsect.dtx v1.0m
General: TO: Promoted documentation to doc.sty standard . . 282
1995/05/06 ltsect.dtx 1.0n
`\@seccntformat`: Use `\quad` instead of `\hskip` . . . 286
`\@sect`: Added `\relax` after `\@seccntformat` just in case . . . 285
1995/05/07 ltboxes.dtx v1.0t
General: Use `\hb@xt@` . . . 233
1995/05/07 ltdefns.dtx v1.2k
`\hb@xt@`: Macro added . . . 23
1995/05/07 ltmath.dtx v1.0r
General: Use `\hb@xt@` . . . 210
1995/05/07 ltoutput.dtx v1.1m
General: Use `\hb@xt@`. . . . 314
1995/05/07 ltpictur.dtx v1.0g
General: Use `\hb@xt@` . . . 260
1995/05/07 ltplain.dtx v1.1j
General: Use `\hb@xt@` . . . 11
1995/05/07 ltsect.dtx v1.0o
General: Use `\hb@xt@` . . . 282
1995/05/07 lttab.dtx v1.0l
General: Use `\hb@xt@` . . . 241
1995/05/08 ltbibl.dtx v1.1g
`\@citex`: Use `\@firstofone` . . . 308
`\bibitem`: Removed unnecessary braces . . . 308
`\nocite`: Use `\@firstofone` . . . 309
1995/05/08 ltdefns.dtx v1.2k
`\typein`: Use `\@firstofone` . . . 24
1995/05/08 ltdefns.dtx v1.2l
`\typein`: Remove unnecessary braces . . . 24
Replace `\def` by `\let` . . . 24
1995/05/08 ltfsstrc.dtx v2.3n
`\ifnot@nil`: Use `\@firstofone` . 135
1995/05/11 fontdef.dtx v2.2j
General: Updates to some plain macros . . . 173
1995/05/12 ltclass.dtx v1.0n
`\DeclareOption*`: Use `\toks@` to remove need to double hash /1557 . . . 368
1995/05/12 ltfloat.dtx v1.1h
`\@footnotemark`: Add `\nobreak` to allow hyphenation. latex/1605 304
1995/05/12 ltpictur.dtx v1.0h
`\pictur@`: Macro added for latex/1355 . . . 261
1995/05/12 ltvers.dtx v1.0e
General: Add autoload docstrip guards . . . 21
Check for format older than 1 year . . . 21
1995/05/13 ltfsstrc.dtx v2.3o
General: Use single hash mark in `\DeclareOption` . . . 126
1995/05/16 ltfloat.dtx v1.1i
`\@makefnmark`: Now use `\textsuperscript`. . . . 303
`\textsuperscript`: Command added./pr1503 . . . 303
`\thefootnote`: Streamlined parts of code. . . . 303

1995/05/17 ltboxes.dtx v1.0u
- `\@irsbox`: Removed surplus braces 240

1995/05/17 ltclass.dtx v1.0o
- `\g@addto@macro`: Make long for latex/1522 374

1995/05/17 ltlists.dtx v1.0g
- `\@item`: Removed surplus braces 229
- `\@nbitem`: Removed surplus braces 229
- `enumerate`: Use `\thr@@` and remove surplus braces 230
- `itemize`: Use `\thr@@` 231

1995/05/18 ltfloat.dtx v1.1j
- `\@makefnmark`: Added `\normalfont`. 303
- `\thempfootnote`: Added `\itshape`. 303

1995/05/19 ltpictur.dtx v1.1a
- General: Support autoloading feature 260

1995/05/20 ltcounts.dtx v1.1b
- `\@definecounter`: Streamlined code 104
- `\@fnsymbol`: Allowing both text and math 105
- `\fnsymbol`: Streamlined code 105

1995/05/20 ltcounts.dtx v1.1c
- `\@definecounter`: And do it right 104

1995/05/20 ltfloat.dtx v1.1k
- `\@makefnmark`: Moved `\normalfont` back and use `\@textsuperscript` 303
  - Moved `\normalfont` to `\textsuperscript` 303
- `\textsuperscript`: Use `\normalfont`. 303

1995/05/21 ltfssdcl.dtx v2.1t
- `\DeclareMathRadical`: Allow for undefined cs names 164

1995/05/21 ltlists.dtx v1.0f
- General: Moved to doc.sty standard 219

1995/05/21 ltmath.dtx v1.0r
- `\@sqrt`: Use `\sqrtsign` 215
- General: Remove `\mathhexbox` from this file 213
  - Update some plain macros 210
- `\lefteqn`: Use `\rlap` 216
- `\r@@t`: Use `\sqrtsign` instead of `\sqrt` 211

1995/05/21 ltoutenc.dtx v1.7h
- `\@inmathwarn`: Added several `\@onlypreamble` 73

1995/05/21 ltoutenc.dtx v1.7j
- General: Updated some plain macros 81

1995/05/21 ltplain.dtx v1.1j
- General: Moved some code to other files 11

1995/05/22 ltplain.dtx v1.1k
- General: Definitions of `\footins` and `\footnoterule` moved to ltfloat. 20

1995/05/22 lttab.dtx v1.1a
- General: Support autoloading feature 241

1995/05/23 ltfssini.dtx v2.2e
- `\newfont`: Font assignment made local again. 169

1995/05/24 ltdefns.dtx v1.1l
- `\newif`: (DPC) New implementation 28

1995/05/24 ltdefns.dtx v1.2m
- `\typein`: (DPC) New implementation 24

1995/05/24 ltfloat.dtx v1.1l
- `\@textsuperscript`: Command added. 303
- General: Moved definition of `\footins` and `\footnoterule` from ltplain. 303
- `\textsuperscript`: Use `\@textsuperscript` 303

1995/05/24 ltfssbas.dtx v3.0a
- General: (DPC) Make file from previous file, fam.dtx 1995/05/20 v2.2d 107
- `\mathgroup`: (DPC) No need to redefine `\newfam` as not outer 108

1995/05/24 ltfsscmp.dtx v3.0a
- General: (DPC) Make file from previous file, fam.dtx 1995/05/20 v2.2d 145

1995/05/24 ltfssdcl.dtx v3.0a
- General: (DPC) Make file from previous file, latint.dtx 1995/05/21 v2.1t 149

1995/05/24 ltfssini.dtx v3.0a
- General: (DPC) Make file from previous file, lfonts.dtx 1995/05/23 v2.2e 168
- `\cal`: (DPC) Remove definition 172
- `\mit`: (DPC) Remove definition 172

1995/05/24 ltfsstrc.dtx v3.0a
- General: (DPC) Make file from previous file, tracefnt 1995/05/16 v2.3o 125

1995/05/24 ltfsstrc.dtx v3.0b
- General: (DPC) Fix `\ProvidesFile` usage 125

1995/05/25 ltclass.dtx v1.0p
- `\endfilecontents`: Delete `\filec@ntents` after preamble 375

1995/05/25 ltfiles.dtx v1.0s
- `\document`: Added check for `\topskip` zero 63

1995/05/25 ltfiles.dtx v1.0t
- `\@iffileonpath`: (CAR) added `\long` 65
- `\document`: Corrected typo 63
- `\IfFileExists`: (CAR) added `\long` 65
- `\InputIfFileExists`: (CAR) added `\long` 66
- `\nofiles`: (CAR) added `\long` 63

- `\protected@write`: (CAR) added `\long` 64

1995/05/25 ltfloat.dtx v1.1m

- `\@savemarbox`: (CAR) Resettings moved to hook 299
- `\@xfloat`: (CAR) Resettings moved to hook 295

1995/05/25 ltlists.dtx v1.0i

- `\endtrivlist`: Macros moved from ltspace.dtx 227

1995/05/25 ltmath.dtx v1.3c classes

- `\@eqnnum`: replace `\reset@font\rmfamily` with `\normalfont` (PR 1578) 216

1995/05/25 ltspace.dtx v1.2f

- `\@vbsphack`: (CAR) not used so 're-moved'. 54
- `\@vspacer`: (CAR) `\@restorepar` added to avoid possible infinite tail recursion caused by a typo in the argument. 55
  - (CAR) macros modified to be more efficient 55
- General: Macros moved to ltlists.dtx 48

1995/05/26 ltdefns.dtx v1.2n

- `\@gobblefour`: (CAR) Added `\longs` 29

1995/05/26 ltmath.dtx v1.0s

- `\@eqnnum`: Removed `\rmfamily` (PR 1578), replaced `\reset@font` with `\normalfont` 214

1995/05/26 ltpage.dtx v1.0g

- `\ps@plain`: removed `\rmfamily` (PR 1578) 312

1995/05/27 ltfssbas.dtx v3.0b

- `\mathgroup`: (FMi) But a need to define `\new@mathgroup` 108

1995/06/05 fontdef.dtx v2.2k

- General: Moved math commands from ltoutenc.dtx. 186

1995/06/05 ltfinal.dtx v1.0r

- General: Added `\MakeUppercase` and `\MakeLowercase`. 379

1995/06/05 ltoutenc.dtx v1.7k

- `\@inmathwarn`: Removed `\protected@cmd` and replaced with explicit `\noexpand`. 73
- General: Allowed `\ProvideTextCommandDefault` after the preamble. 74
  - Commented out `\textless` and `\textgreater`. 80
  - Moved math commands to fontdef.dtx. 81
  - Save some tokens in `\textvisiblespace` and `\textunderscore`. 80

1995/06/06 ltfinal.dtx v1.0s

- General: Made `\MakeUppercase` and `\MakeLowercase` brace their argument. 379

1995/06/09 ltoutenc.dtx v1.7l

- `\DeclareTextComposite`: Rewrote `\DeclareTextComposite` to define the composite as a no-argument command rather than a two-argument command. 75

1995/06/11 ltspace.dtx v1.2g

- `\restorecr`: (CAR) `\relax` added to stop silent eating of *. 58

1995/06/13 ltfinal.dtx v1.0t

- General: Add patch level string more carefully 384
  - Call `\errorstopmode` 385

1995/06/13 ltpictur.dtx v1.1b

- General: Use `\ProvidesFile` in autoload 260

1995/06/14 lttab.dtx v1.1b

- General: Use `\ProvidesFile` in autoload 241

1995/06/15 ltfssbas.dtx v3.0c

- General: (DPC) minor documentation changes 107

1995/06/15 ltfsscmp.dtx v3.0b

- General: (DPC) minor documentation edits 145

1995/06/15 ltfssdcl.dtx v3.0b

- General: (DPC) minor documentation changes 149

1995/06/19 ltbibl.dtx v1.1h

- `\bibcite`: Call `\@newl@bel` so repeated keys produce better warning. 308

1995/06/19 ltclass.dtx v1.0q

- `\documentclass`: Dont redefine `\usepackage` in compat mode for /1634 370

1995/06/19 ltxref.dtx v1.1e

- `\newlabel`: Use `\@newl@bel` to share code with `\bibcite` 200

1995/06/28 ltfssini.dtx v3.0b

- General: (DPC) Fix documentation typos 168

1995/06/28 ltmath.dtx v1.0t

- General: minor doc edits 210

1995/07/02 ltplain.dtx v1.1n

- General: Removed surplus 'by' and '=' in various places 11
- `\offinterlineskip`: Replaced 1000 by `\@m` 18
- `\showoutput`: Use `\showoverfull` to save space 20
- `\tracingall`: Use `\showoutput` to save space 20

1995/07/03 ltdefns.dtx v1.2o

- `\set@typeset@protect`: Use `\@typeset@protect` for init 31

1995/07/03 ltfntcmd.dtx v3.3s

- `\t@st@ic`: Use clean interface for jump 195

1995/07/05 ltfntcmd.dtx v3.3s

- `\t@st@ic`: Renamed from `\test@next` 195

- 1995/07/05 ltspace.dtx v1.2h
  - \@gnewline: Use \break 52
  - \@no@pgbk: Macro replaces \@pgbk and \@nopgbk 51
  - \nopagebreak: Reimplemented both using \@no@pgbk 50
- 1995/07/09 ltcntrl.dtx v1.0f
  - \@iforloop: Reimplemented using Kabelschacht method 37
  - \@iwhiledim: Reimplemented using Kabelschacht method 36
  - \@iwhilenum: Reimplemented using Kabelschacht method 36
  - \@iwhilesw: Reimplemented using Kabelschacht method 36
  - \@tfor: Reimplemented using Kabelschacht method 37
- 1995/07/09 ltlists.dtx v1.0j
  - enumerate: Use \expandafter . . 230
  - itemize: Use \expandafter 231
- 1995/07/12 ltpictur.dtx v1.1d
  - General: allow 2e commands in 209 mode. latex/1737 260
- 1995/07/13 ltdefns.dtx v1.0p
  - General: Updates to documentation 22
- 1995/07/13 ltfiles.dtx v1.0u
  - General: Updates to docu 60
- 1995/07/13 ltfssbas.dtx v3.0d
  - \@@defaultsubs: macro added . . 120
  - \@defaultsubs: macro added . . . 120
  - General: minor documentation changes 107
  - \wrong@fontshape: Change a macro not a switch to flag default font substitutions 119
- 1995/07/13 ltmiscen.dtx v1.0z
  - \@centercr: Use \nobreak 206
  - \@writefile: Added missing percent and use \relax in the THEN case 204
  - \@xobeysp: Use \nobreak 207
  - General: Improve Documentation 202
  - \enddocument: Set \@setckpt to \@gobbletwo instead of defining it by hand 203
    - Shorten redefinition of \bibcite and \newlabel 203
    - Use \@defaultsubs instead of switch 204
- 1995/07/14 ltbibl.dtx v1.1i
  - \bibcite: Remove \@onlypreamble so still defined in new \enddocument 308
- 1995/07/14 ltxref.dtx v1.1g
  - \newlabel: Remove \@onlypreamble so still defined in new \enddocument 200
- 1995/07/19 ltfssini.dtx v3.0d
  - General: (DPC) TeX2 support . . 171
- 1995/07/20 ltboxes.dtx v1.0v
  - \@isavebox: Use \sbox 235
  - \@isavepicbox: Use \sbox 235
- 1995/07/21 ltoutput.dtx v1.1o
  - \@writesetup: Command added . 335
    - New, experimental, versions: need in-lining 335
- 1995/08/09 ltmath.dtx v1.0u
  - General: Added code for class options leqno and fleqn 216
- 1995/08/11 ltlength.dtx v1.1b
  - General: Doc typos fixed for latex/753 106
- 1995/08/16 ltcntrl.dtx v1.0g
  - \@break@tfor: Made long 37
  - \@forloop: Made defs long 37
  - \@fornoop: Made defs long 37
  - \@iforloop: Made defs long 37
  - \@iwhiledim: Made defs long . . . 36
    - Removed \@whilenoop 36
  - \@iwhilenum: Made defs long . . . 36
    - Removed \@whilenoop 36
  - \@iwhilesw: Removed \@whileswnoop 36
  - \@tfor: Made defs long 37
- 1995/08/16 ltfiles.dtx v1.0v
  - \document: set \@maxdepth 63
    - set \do globally 63
    - set \topskip globally 63
- 1995/08/24 ltfssbas.dtx v3.0f
  - General: Added autoload code . . 107
- 1995/08/24 ltfsstrc.dtx v3.0c
  - General: Macro \gobble@font@spec removed 135
  - \tryis@simple: 141
- 1995/08/25 ltoutput.dtx v1.1p
  - General: Support autoloading feature (FMi). 314
- 1995/09/01 lterror.dtx v1.2i
  - General: Add autoload support . 38
- 1995/09/01 ltplain.dtx v1.1m
  - \empty: Use \let to save space . 17
  - \I: Use \let to save space 17
- 1995/09/14 ltplain.dtx v1.1o
  - General: Moved \multispan to lttab.dtx 11
- 1995/09/14 lttab.dtx v1.1c
  - \cline: (DPC) New implementation 259
- 1995/09/15 ltfssini.dtx v3.0e
  - General: (DPC) Modify TeX2 message 171
- 1995/09/19 ltmiscen.dtx v1.1a
  - \verb: Put \@noligs after \verbatim@font where it belongs. . 208
- 1995/10/01 ltfiles.dtx LaTeX2e
  - \@addtofilelist: Macro added . 67
- 1995/10/02 ltdefns.dtx v1.2q
  - \@autoload: Macro added 33
  - \@if: Add \aut@global in autoload version 28
  - \@ifnch: Use \@let@token for internal/924, save \reserved@e 32
  - \@ifnextchar: Use \@let@token 32

- `\@newenv`: Add `\aut@global` in autoload version . . . 28
- `\@protected@testopt`: Macro added . . . 25
- `\@testopt`: Macro added . . . 25
- `\@xargdef`: Add `\aut@global` in autoload version . . . 25
  - New implementation, using `\@test@opt` . . . 25
- `\@yargd@f`: Add `\aut@global` in autoload version . . . 26
- `\aut@global`: Macro added . . . 33
- `\newif`: Add `\aut@global` in autoload version . . . 28
- `\renew@environment`: Add `\aut@global` in autoload version . . . 27

1995/10/02 ltplain.dtx v1.1p
- General: Move `\newif` to ltdefns . . . 14

1995/10/03 fontdef.dtx v2.2l
- General: `\@@sqrt` from patch file for /1701 . . . 173

1995/10/03 ltdefns.dtx v1.2r
- `\typein`: Add missing `\@typein` for /1710 (from patch file) . . . 24

1995/10/03 ltpictur.dtx v1.1e
- General: New autoload code . . . 260

1995/10/04 ltfssbas.dtx v3.0g
- General: Modify autoload code . . . 107

1995/10/04 ltfsstrc.dtx v3.0d
- General: (DPC) Modify autoload code . . . 125

1995/10/04 lttab.dtx v1.1d
- General: Modify autoload support 241

1995/10/06 ltdefns.dtx v1.2s
- `\declare@robustcommand`: Add `\aut@global` in autoload version . . . 31

1995/10/06 ltfiles.dtx v1.0w
- `\@missingfileerror`: Autoload error . . . 66

1995/10/09 ltdefns.dtx v1.2t
- `\@autoload`: Use `\@@input` not `\input` to save string space and stops autoload files appearing in `\listfiles` . . . 33

1995/10/09 lterror.dtx v1.2j
- General: Modify autoload support 38

1995/10/09 ltoutenc.dtx v1.7m
- `\@inmathwarn`: Autoload error . . . 74

1995/10/10 ltfssbas.dtx v3.0h
- `\showhyphens`: Use `\normalfont` and make colour safe, and autoloadable . . . 123

1995/10/10 ltfssdcl.dtx v3.0c
- `\non@alpherr`: (DPC) autoload error message . . . 152

1995/10/10 ltplain.dtx v1.1r
- General: Autoload tracing code . . . 11

1995/10/10 ltthm.dtx v1.0f
- General: Make `\newtheorem` ‘only preamble’ . . . 279

1995/10/11 ltoutput.dtx v1.1r
- `\clearpage`: Added a check so that it does not lose the argument of `\twocolumn[...]` . . . 325

1995/10/16 ltbibl.dtx v1.1j
- `\cite`: (DPC) Make robust . . . 308

1995/10/16 ltboxes.dtx v1.0w
- General: Clarify makebox description . . . 233

1995/10/16 ltdefns.dtx v1.2u
- `\@ifstar`: (DPC) New implementation, for /1910 . . . 33
- `\new@command`: (DPC) Use `\@testopt` /1911 . . . 25
- `\new@environment`: (DPC) Use `\@testopt` /1911 . . . 27
- `\typein`: (DPC) Use `\@testopt` /1911 . . . 24

1995/10/16 ltfssini.dtx v3.0f
- `\p@reset@font`: Added `\relax` after `\usefont`, as the latter eats up spaces. . . . 170

1995/10/16 ltmath.dtx v1.0y
- `\@yeqncr`: (DPC) Use `\@testopt` /1911 . . . 215
- `\sqrt`: (DPC) Make robust /1808 215

1995/10/16 ltspace.dtx v1.2j
- `\nolinebreak`: (DPC) Use `\@testopt` /1911 . . . 51
- `\nopagebreak`: (DPC) Use `\@testopt` /1911 . . . 50

1995/10/16 ltthm.dtx v1.0g
- General: Revert to previous `\newtheorem` behaviour . . . 279

1995/10/17 ltclass.dtx v1.0r
- `\@providesfile`: Delay definition of `\ProvidesFile` till ltfinal 368
- `\ProcessOptions*`: Reset `\CurrentOption` for graphics/1873 . . . 370

1995/10/17 ltdirchk.dtx v1.0l
- General: Modify initex version of `\ProvidesFile` . . . 3

1995/10/17 ltfinal.dtx v1.0v
- `\@providesfile`: reset macro . . . 385
- `\reserved@b`: reset here after the `\input` above . . . 384

1995/10/17 ltplain.dtx v1.1s
- `\eject`: Move `\supereject` to compat file . . . 18

1995/10/17 lttab.dtx v1.1e
- `\@cline`: (DPC) Use `\@multicnt` 259
- `\@multispan`: (DPC) Macro added. . . . 259

1995/10/19 ltfinal.dtx v1.0w
- `\@filelist`: Move after `\reserved@a` setting:-) . . . 385

1995/10/20 ltbibl.dtx v1.1k
- `\@citex`: Removed refundefined flag . . . 308
- `\nocite`: Removed refundefined flag . . . 309

1995/10/20 ltclass.dtx v1.0s
- `\@begindocumenthook`: Make setting conditional, for autoload version . . . 374

1995/10/20 ltfssbas.dtx v3.0i
- General: (DPC) Modify autoload code, change `\undefined` . . . 107

1995/10/20 ltfsstrc.dtx v3.0e
- General: (DPC) Modify autoload code . . . 125

1995/10/22 ltfssbas.dtx v3.0j
- General: (RmS) New size function macro `\genb@sfcnt` needs to be disabled at `\document`. . . . 107

1995/10/22 ltfsstrc.dtx v3.0f
- General: Added 'genb' and 'sgenb' size functions to support new DC font naming scheme. . . . 125

1995/10/23 lttab.dtx v1.1f
- `\@settab`: (CAR)Ensure that `\@hightab` increases by at most one . . . 247
- `\@startline`: (CAR)Ensure that `\@nxttabmar` is never larger than `\@hightab` . . . 245
- `\poptabs`: (CAR)Ensure that `\@curtab` is never larger than `\@hightab` . . . 248
- `\tabbing`: (CAR)Make `\@hightab` consistently a local variable . . . 246

1995/10/24 ltdefns.dtx v1.2v
- `\@autoload`: ignore end-of-line . . . 33

1995/10/24 lterror.dtx v1.2k
- `\@preamerr`: Modify autoload support . . . 44

1995/10/24 ltfiles.dtx v1.1a
- `\document`: Removed multiplelabels switch . . . 62
  - Removed refundefined switch . . . 63

1995/10/24 ltfssbas.dtx v3.0k
- `\@@defaultsubs`: macro removed 120
- `\wrong@fontshape`: Make this code inline since it happens only here . . . 119

1995/10/24 ltmiscen.dtx v1.1b
- `\enddocument`: Changed logic for producing warning messages and removed switch . . . 204
  - Use `\@refundefined` instead of switch . . . 204

1995/10/24 ltxref.dtx v1.1h
- `\@multiplelabels`: Switch for multiplelabels removed . . . 200
- `\@newl@bel`: Switch for multiplelabels replaced by inline code . . . 200
- `\@refundefined`: Switch for refundefined replaced . . . 199
- `\@setref`: Switch for refundefined renamed . . . 200
- `\if@multiplelabels`: Macro removed . . . 200

1995/10/25 ltalloc.dtx v1.1b
- General: General doc improvements . . . 34

1995/10/25 ltfloat.dtx v1.1n
- `\@endfloatbox`: (CAR) macro added: to unify code for double and single versions . . . 297
- `\end@dblfloat`: (CAR) unify code for double and single versions 296
- `\end@float`: (CAR) unify code for double and single versions . . . 296

1995/10/25 ltidxglo.dtx v1.1d
- General: Doc cleanup . . . 305

1995/10/25 ltsect.dtx v1.0q
- `\subparagraphmark`: Use `\let` not `\def` to save space. . . . 288

1995/10/26 ltfssbas.dtx v3.0l
- `\define@newfont`: (DPC) disable autofss2 for now . . . 117

1995/10/27 ltpictur.dtx v1.1f
- General: Move initialisation to kernel from autoload file . . . 276

1995/10/31 ltboxes.dtx v1.0x
- `\@finalstrut`: Add `\nobreak` in horiz mode to allow hyphenation. internal/1931 . . . 240

1995/11/01 fontdef.dtx v2.2m
- General: add `\nfss@catcodes` for internal/1932 . . . 175

1995/11/01 ltdirchk.dtx v1.0n
- General: Initialise `\@addtofilelist` to `\@gobble` . . . 3

1995/11/01 ltfinal.dtx v1.0x
- General: (DPC) Switch meaning of `\@addtofilelist` for cfg files 380

1995/11/01 ltfssbas.dtx v3.0m
- `\DeclareFontShape`: (DPC) Test for `\relax` not `\undefined`, internal/1933 . . . 108

1995/11/01 ltfssini.dtx v3.0g
- General: (DPC) Switch meaning of `\@addtofilelist` for cfg files 171

1995/11/02 ltfssbas.dtx v3.0n
- `\wrong@fontshape`: (DPC) Remove extra space with `\string` for latex/1676 . . . 119

1995/11/02 ltoutenc.dtx v1.7n
- General: Changed internal name `\a` to `\@tabacckludge` to protect against redefinition by malicicous users. . . . 78

1995/11/07 ltlists.dtx v1.0k
- `\@doendpe`: Enclosed `\setbox0` assignment by a group so that it leaves the contents of box 0 intact. . . . 227

1995/11/07 ltoutenc.dtx v1.7o
- General: Added `\leavevmode` at start of `\c`, otherwise the output routine might be invoked within the macro. . . . 81

- Changed `\char32` to `\@xxxii` (two tokens less). 82
- Replaced octal number 27 by decimal number 23 to protect against the quote character being active. 82
- Replaced some 0's by `\z@` (faster). 82

1995/11/10 ltoutput.dtx v1.1s
- `\@shipoutsetup`: Command removed 335
- `\@writesetup`: Command removed 335
  - In-lined 335

1995/11/14 ltclass.dtx v1.0t
- `\@@unprocessedoptions`: Allow empty option 375
- `\@loadwithoptions`: macro added 371
- `\LoadClassWithOptions`: macro added 371
- `\RequirePackageWithOptions`: macro added 371

1995/11/17 ltfssbas.dtx v3.0m
- `\@wrong@font@char`: (DPC) Macro added. latex/1676 120
- `\define@newfont`: Redefine `\typeout` latex/1676 116
- `\wrong@fontshape`: Support `\@wrong@font@char` latex/1676 119

1995/11/17 ltoutenc.dtx v1.7p
- `\UseTextSymbol`: Support `\@wrong@font@char` latex/1676 76

1995/11/18 ltoutenc.dtx v1.7q
- `\UseTextSymbol`: Modify message slightly 76

1995/11/21 fontdef.dtx v2.2n
- General: Incorporate changed figures, as in plain.tex 185

1995/11/27 ltfssbas.dtx v3.0n
- `\nfss@catcodes`: Reset hash, for definitions in fd files 118

1995/11/28 lterror.dtx v1.2l
- `\ClassInfo`: Typo in autoload code /1985 41

1995/11/28 ltfloat.dtx v1.1n
- General: documentation fixes 291

1995/11/28 ltfsstrc.dtx v3.0g
- General: documentation fixes 125

1995/11/28 ltoutenc.dtx v1.7r
- General: Added math mode checks to text commands. 72
  - doc fixes 69
  - Renamed `\@changed@x@err` to `\TextSymbolUnavailable`. 72

1995/11/29 ltoutenc.dtx v1.7t
- General: Added `\textasciicircum`, `\textasciitilde`, `\textbackslash`, `\textbar`, `\textgreater` and `\textless`. 83
  - Added `\textasciicircum`, `\textasciitilde`, `\textregistered` and `\texttrademark`. 80
  - Added `\textbackslash` and `\textbar`. 79, 86
  - Added `\textless` and `\textgreater`. 80, 87

1995/12/01 ltoutenc.dtx v1.7u
- General: Made `\SS` a Default, rather than having the default point to the OT1 definition. 80

1995/12/04 ltspace.dtx v1.2k
- `\nobreakspace`: (Macro added 57

1995/12/04 ltspace.dtx v1.2l
- `\@xobeysp`: (braces added to definition of tilde 57

1995/12/04 preload.dtx v2.4e
- General: Ulrik Vieth. added 12pt OMS and OML preloads /1989 190

1995/12/05 ltdefns.dtx 1.2w
- `\@unexpandable@noexpand`: Removed as never used. internal/1733 30

1995/12/05 ltfiles.dtx v1.1c
- `\document`: `\ignorespaces` added for latex/1933 63

1995/12/05 ltfloat.dtx v1.1n
- `\@textsuperscript`: Use `\ensuremath` for latex/1984. 303

1995/12/05 ltoutenc.dtx v1.7v
- `\@inmathwarn`: Changed `\TextSymbolUnavailable` text 74

1995/12/06 ltfssbas.dtx v3.00
- `\nfss@catcodes`: Reset hat, for typeouts etc in fd files 118

1995/12/07 ltbibl.dtx v1.1l
- `\@citex`: Restored name of `\G@refundefinedtrue` 308

1995/12/07 ltfloat.dtx v1.1m
- `\@textsuperscript`: Move `\m@th` out of the `\ensuremath` for latex/1984. 303

1995/12/07 ltxref.dtx v1.1i
- `\@setref`: Switch for refundefined restored 200
- `\G@refundefinedtrue`: Renamed (back) from `\G@refundefined` 199

1995/12/11 ltoutenc.dtx v1.7w
- General: Modified `\copyright` 80

1995/12/13 ltdefns.dtx 1.2x
- `\-`: Documentation changed. 22

1996/01/10 ltfiles.dtx v1.1d
- `\@iffileonpath`: Change argument handling to not require doubled hash. latex/2024 65

1996/01/20 ltidxglo.dtx v1.1e
- `\makeglossary`: Make no-op after use pr/2048 306
- `\makeindex`: Make no-op after use pr/2048 306

1996/01/20 ltspace.dtx v1.2m
- `\vspace`: Made robust 55

1996/03/25 ltmath.dtx v1.1a
- `\@ensuredmath`: Macro added for amslatex/2104 216

  `\ensuremath`: Reimplement for amslatex/2104 216
1996/04/18 ltpage.dtx v1.0i
  General: Improve documentation 311
1996/04/22 ltmiscen.dtx v1.1c
  General: Improve Documentation 202
1996/04/22 ltspace.dtx v1.2n
  General: Documentation Improvements 48
1996/04/22 lttab.dtx v1.1g
  `\@tabclassz`: (DPC) Extra `\hskip` keeps tabcolsep in empty columns internal/2122 257
1996/04/23 ltcounts.dtx v1.1d
  General: Documentation improvements 103
1996/04/24 ltfiles.dtx v1.1e
  `\document`: (DPC) Reset `\AtBeginDocument` eg for latex/1297 63
1996/05/08 ltfsstrc.dtx v3.0h
  `\math@egroup`: Use `\bgroup` instead of `\begingroup` to match a kernel change made in 1994!! 134
1996/05/09 ltfntcmd.dtx v3.3t
  `\check@icr`: Default definitions added 194
1996/05/17 fontdef.dtx v2.2o
  General: `\@@sqrt` removed, at last 173, 185
1996/05/17 ltfiles.dtx v1.1f
  `\nofiles`: added `\write` to `\protected@write` for latex/2146 63
1996/05/18 ltoutenc.dtx v1.7x
  General: Produce error if encoding not found. pr/2054 93
1996/05/21 ltoutenc.dtx v1.7y
  General: Corrected error message (CAR) 93
1996/05/21 ltsect.dtx v1.0s
  `\@sect`: (DPC) Added extra braces for internal/2148 285
  (DPC) Moved brace to allow commands like `\MakeUppercase` in 6th argument. Changed `\par` to `\endgraf` to allow non-long commands. internal/2148 285
  `\@ssect`: (DPC) Added extra braces for internal/2148 287
  (DPC) Moved brace to allow commands like `\MakeUppercase` in 4th argument. Changed `\par` to `\endgraf` to allow non-long commands. internal/2148 287
1996/05/23 ltoutenc.dtx v1.7z
  `\@strip@args`: `\expandafter` added to match other changes for latex/2133 76
  `\add@accent`: macro added. latex/2133 74
  `\DeclareTextAccent`: Reimplemented using `\add@accent` to save space latex/2133 74
  `\DeclareTextCompositeCommand`: Modified to cope with new `\add@accent` command: required removal of check for one argument-command 75
1996/05/24 ltoutput.dtx v1.1t
  `\@specialoutput`: Check that `\@colroom` is less than `\vsize`, indicating that a float has been added 328
  Cut-off point changed to 1.5`\baselineskip` 328
  `\@topnewpage`: Cut-off point changed to 2.5`\baselineskip` 327
1996/05/25 ltoutput.dtx v1.1u
  `\@specialoutput`: Correct the above check 328
1996/06/03 ltmiscen.dtx v1.1d
  `\@verbatim`: Exchanged the following two code lines so that `\dospecials` cannot reset the category code of characters handled by `\@noligs`. 207
  General: Move setting of verbatim font and `\@noligs`. 202
  `\verb`: Put setting of verbatim font after `\dospecials` so that `\dospecials` cannot reset the category code of characters handled by `\@noligs`. 208
1996/06/10 ltboxes.dtx v1.0y
  `\@parboxto`: (DPC) Changed `\endgraf` to `\@@par` 237
1996/06/10 ltsect.dtx v1.0t
  `\@sect`: (DPC) Changed `\endgraf` to `\@@par` 285
  `\@ssect`: (DPC) Changed `\endgraf` to `\@@par` 287
1996/06/13 ltdirchk.dtx v1.0r
  General: documentation improvements mainly from internal/2174 1
1996/06/14 lttab.dtx v1.1h
  `\@tabclassz`: (DPC) Change both`\z@skip` to 1sp for latex/2160 257
1996/06/22 ltspace.dtx v1.2o
  General: Documentation of problems added 48
1996/07/10 ltfinal.dtx v1.0y
  `\toks`: Free up memory from scratch registers /2213 384
1996/07/19 ltoutenc.dtx v1.8a
  `\@strip@args`: Use char 0 not @ as carrier for `\lowercase` /2197 76
1996/07/26 ltboxes.dtx v1.0z
  `\if@minipage`: put `\global` into definition 238
1996/07/26 ltclass.dtx v1.0u
  `\@classoptionslist`: made only preamble 366

  - `\@unusedoptionlist`: made only preamble 366
- 1996/07/26 ltdefns.dtx v1.2y
  - `\@reargdef`: third arg picked up by `\@yargdef` 26
  - `\renew@command`: use `\noexpand` instead of `\string` 26
    - use `\relax` in place of empty arg 26
  - `\renew@environment`: use `\relax` in place of empty arg 27
- 1996/07/26 ltfloat.dtx v1.1n
  - `\@endfloatbox`: remove unecessary `\global` before `\@minipage`... 297
  - `\@savemarbox`: remove unecessary `\global` before `\@minipage`... 299
  - `\@setminipage`: remove unecessary `\global` before `\@minipage`... 296
  - `\@setnobreak`: remove unecessary `\global` before `\@nobreak`... 296
- 1996/07/26 ltfssbas.dtx v3.0p
  - `\@DeclareMathSizes`: use faster `\if` test 112
  - `\nfss@catcodes`: omit `\relax` as not needed 118
- 1996/07/26 ltfssdcl.dtx v3.0e
  - `\init@restore@version`: Removed `\ifrestore@version` switch and replaced by `\init@restore@version` 151
- 1996/07/26 ltfsstrc.dtx v3.0i
  - `\init@restore@glb@settings`: macro added replacing `\if@inmath` switch 133
- 1996/07/26 ltlists.dtx v1.0l
  - `\@item`: Remove unecessary `\global` before `\@minipage`... 228
    - Remove unecessary `\global` before `\@nobreak`... 229
- 1996/07/26 ltmath.dtx v1.1b
  - General: Removed `\global` before `\@ignoretrue` in various places. 210
- 1996/07/26 ltmiscen.dtx v1.1e
  - `\@ignorefalse`: put `\global` into definition 203
  - `\begin`: remove `\global` before `\@ignore`... 205
  - `\end`: remove `\global` before `\@ignore`... 205
  - `\ignorespacesafterend`: user level macro added 203
- 1996/07/26 ltoutput.dtx v1.1v
  - `\@testfp`: remove `\global` before `\@test`... 353
  - `\@xtryfc`: remove `\global` before `\@test`... 340
  - `\@ztryfc`: remove `\global` before `\@test`... 341
  - `\clearpage`: add number of missing percents 325
- 1996/07/26 ltplain.dtx v1.1t
  - `\sh@ft`: replace `\dimen\z@` by `\dimen@` 19
- 1996/07/26 ltsect.dtx v1.0u
  - `\@starttoc`: removed `\global` before `\@nobreak`... 288
  - `\@xsect`: Removed `\global` before `\@nobreak`... 286
- 1996/07/26 ltspace.dtx v1.2p
  - `\if@nobreak`: put `\global` inside definition 52
- 1996/07/27 ltfssbas.dtx v3.0q
  - General: `\if@inmath` switch removed 115
- 1996/07/27 ltspace.dtx v1.2q
  - General: Further documentation of problems 48
- 1996/07/27 ltspace.dtx v1.2r
  - General: Correct documentation of problems 48
- 1996/08/02 ltfloat.dtx v1.1o
  - `\@xympar`: Remove `\global` before `\@ignore`... 300
- 1996/08/02 ltsect.dtx v1.0v
  - `\@afterheading`: Removed `\global` before `\@nobreak`... 287
- 1996/08/02 ltspace.dtx v1.2s
  - `\@Esphack`: Remove `\global` before `\@ignore`... 53
- 1996/08/25 ltfssbas.dtx v3.0r
  - `\nfss@catcodes`: Reset the acute, grave and double quote chars as well 118
- 1996/09/21 ltoutput.dtx v1.1w
  - `\@writesetup`: Added `\@parboxrestore` and made consequent deletions: wait for the howls of protest 335
- 1996/09/25 ltdirchk.dtx v1.0t
  - General: Move ltxcheck to separate file 10
- 1996/09/28 ltmiscen.dtx v1.1f
  - `\@xobeysp`: Moved to ltspace.dtx 207
- 1996/09/28 ltspace.dtx v1.2t
  - `\@xobeysp`: Moved from ltmiscen.dtx and redefined to use `\nobreakspace` 57
- 1996/09/29 ltfiles.dtx v1.1g
  - `\document`: Added disabling of `\@nodocument` 63
- 1996/09/29 ltoutput.dtx v1.1x
  - `\newpage`: Checks for noskipsec and inlabel added 326
- 1996/09/29 ltsect.dtx 1.0w
  - `\@noskipsectrue`: Added documentation 283
- 1996/09/30 ltoutput.dtx v1.1y
  - `\newpage`: Checks for noskipsec and inlabel removed pending further tests 326

1996/10/04 ltclass.dtx v1.0v
  `\RequirePackageWithOptions`: Reset `\@unprocessedoptions` for /2269 . . . 371

1996/10/05 ltfiles.dtx v1.1h
  `\@clubpenalty`: Added setting its value . . . 62

1996/10/08 ltfntcmd.dtx v3.3u
  `\DeclareTextFontCommand`: Removed `\check@icr` when in vmode since it causes various errors (see pr/2157) . . . 193

1996/10/21 lttab.dtx v1.1i
  `\@array`: Use `\set@typeset@protect` . . . 251
  General: Moved the code associated with `\@mkpream` into the group provided by the box, for robustness (latex/2183) . . . 251
  `\multicolumn`: Make `\multicolumn` long (latex/2180) . . . 253
  `\tabbing`: Moved the `\indent` so that the `\everypar` can remove it when necessary; this is needed because the code for items in lists has changed (see pr/22111) . . . 246

1996/10/23 ltlists.dtx v1.0m
  `\@item`: `\@nobreak...` moved into the `\everypar` and not executed unconditionally, see above . . . 229
    `\kern...` changed to `\setbox...` . . . 228
    Added setting of `\clubpenalty` and set `\@nobreakfalse` only when necessary . . . 229

1996/10/23 ltsect.dtx v1.0x
  `\@xsect`: Replaced `\hskip...` with `\setbox...` as used in `\@afterheading` . . . 286

1996/10/24 ltboxes.dtx v1.1a
  `\@arrayparboxrestore`: Added local settings of flags: dangerous! 238
  `\@iiiminipage`: Use it or lose it (@setminpage): Frank will want to lose it . . . 238

1996/10/24 ltfloat.dtx v1.1p
  `\@floatboxreset`: Added local settings of flags: dangerous! . . . 295
  `\@marginparreset`: Added local settings of flags: dangerous! . 300
  `\@xfloat`: Added `\@nodocument` to trap floats in the preamble . 294

1996/10/24 ltoutput.dtx v1.1z
  `\@addtocurcol`: Added `\nobreak`, etc as appropriate . . . 344
  `\@specialoutput`: Added `\nobreak` as appropriate . . . 330
  `\@topnewpage`: Added `\@nodocument` to trap `\twocolumn` in the preamble . . . 326
  `\newpage`: Better checks for noskipsec and inlabel added, plus nobreak . . . 326

1996/10/25 ltlists.dtx v1.0n
  `\endtrivlist`: Change `\indent` to `\leavevmode` . . . 227
    Reset flags explicitly . . . 227

1996/10/25 ltoutput.dtx v1.2a
  `\newpage`: Reset all flags explicitly 326

1996/10/26 ltlists.dtx v1.0o
  `\endtrivlist`: Correct typo . . . 227

1996/10/27 ltoutenc.dtx v1.8c
  `\@strip@args`: Removed macro . 75
  General: Added `\r` A . . . 82
    Added `\textasteriskcentered` . . . 79, 86
    Corrected syntax descriptions . 69
    Removed `\aa` and `\AA` . 79, 82, 83

1996/10/28 ltplain.dtx v1.1u
  General: (CAR) More doc changes 11
  `\dotfill`: Removed math mode . 20

1996/10/29 ltplain.dtx v1.1v
  `\dotfill`: Got arithmetic correct (CAR) . . . 20

1996/10/29 ltspace.dtx v1.2u
  `\@gnewline`: Added macro . . . 52
  `\@no@lnbk`: Macro replaces `\@lnbk` and `\@nolnbk` . . . 51
  `\\`: Corrected and rationalised code 51
  `\nolinebreak`: Reimplemented both using `\@no@lnbk` . . . 51

1996/10/31 ltfinal.dtx v1.0z
  General: Added extra `\lcode`, hoping it does no harm in T1 (pr/1969) . . . 380, 383

1996/10/31 ltlists.dtx v1.0p
  `\@trivlist`: Added check for missing item in outer list . . . 226

1996/10/31 ltsect.dtx v1.0y
  General: Corrected and tidied documentation; removed long lines 282

1996/11/03 ltplain.dtx v1.1w
  `\dotfill`: Saved tokens by using `\hb@xt@` . . . 20

1996/11/04 lterror.dtx v1.2m
  `\@nodocument`: Always define `\@nodocument` in kernel, so that it can be cleared by `\document`. 43

1996/11/04 ltlists.dtx v1.0q
  `\@trivlist`: Moved check for missing item: only checked when not inlabel flag is false . . . 226

1996/11/05 ltfiles.dtx v1.1i
  `\nofiles`: Standard `\if@nobreak` test added . . . 63

1996/11/09 ltmath.dtx v1.1c
  `\@ensuredmath`: Made long, as it was before. /2104 . . . 216

1996/11/18 ltfssbas.dtx v3.0s
  `\define@newfont`: (DPC) lowercase fd file names. internal/1044 . 117

1996/11/18 ltoutenc.dtx v1.8d
- General: (DPC) lowercase external file names. internal/1044 93

1996/11/20 fontdef.dtx v2.2p
- General: lowercase fd and enc.def file names /1044 173

1996/11/20 ltvers.dtx v1.0f
- General: Check for old format modified /2319 21

1996/11/23 ltoutenc.dtx v1.8e
- General: Corrected description 69, 70
  - Extended description 70

1996/11/28 ltvers.dtx v1.0g
- General: Check for old format modified /2319 21

1996/12/06 ltdirchk.dtx v1.0u
- `\IfFileExists`: *** removed from various messages for GNU Make. internal/2338 7

1996/12/06 ltfloat.dtx v1.1r
- `\@caption`: Call `\@setminpage` if needed. latex/2318 293

1996/12/06 ltfssini.dtx v3.0h
- General: (DPC) Remove *** from messages internal/2338 171

1996/12/17 ltclass.dtx v1.0w
- `\g@addto@macro`: Use `\begingroup` to save making a mathord . . 374

1996/12/20 ltsect.dtx v1.0z
- `\@dottedtocline`: Added `\nobreak` for latex/2343 290

1997/01/08 fontdef.dtx v2.2q
- General: Use `\DeclareMathDelimiter` to set delimiter codes . 179
- `\mathparagraph`: Define using `\DeclareMathSymbol` 186

1997/01/08 ltfiles.dtx v1.1j
- `\@include`: reset `\deadcycles` latex/2365 65

1997/01/08 ltmath.dtx v1.1d
- `\root`: (DPC) Remove spurious space tokens from plain TeX definition /2359 211

1997/02/05 ltclass.dtx v1.0x
- `\g@addto@macro`: missing percent /2402 374

1997/02/21 ltlists.dtx v1.0r
- `\@item`: `\ifvoid` check added for `\noindent`. latex/2414 228

1997/03/21 ltcounts.dtx v1.1e
- `\fnsymbol`: Use `\mathsection` and `\mathparagraph`. latex/2445 105

1997/04/14 ltfiles.dtx v1.1k
- `\document`: Set the document space factor defaults. latex/2404 . . . 62
- `\normalsfcodes`: Macro added (from patch file) latex/2404 . . 63

1997/04/14 ltoutput.dtx v1.2b
- `\@writesetup`: Call `\normalsfcodes` (from patch file) latex/2404 336
  - Move `\label` and `\index` (from patch file) 336

1997/04/24 ltbibl.dtx v1.1m
- `\@citex`: `\@empty` to avoid primitive error on empty cite keys. latex/2432 308

1997/04/30 ltoutenc.dtx v1.9a
- General: Changed `\textsc` to `\scshape` 80
  - Introduced `\textcopyright` and modified `\copyright` 80
  - Introduced `\textcopyright` and modify `\copyright` 80
  - Modified `\textunderscore`, removing `\mathunderscore` . . . 80
  - Modified `\underscore`, removing `\mathunderscore` 80

1997/04/30 ltoutenc.dtx v1.9b
- General: Added `\leavevmode` to `\textunderscore` 80

1997/05/04 ltoutenc.dtx v1.9c
- General: Added 'hex index tabs' . 84
  - Added TS1 encoding v2.2.beta 89

1997/05/07 ltoutenc.dtx v1.9d
- General: Added `\leavevmode` to `\textcompwordmark` 80

1997/05/07 ltspace.dtx v1.2v
- `\newline`: Made completely robust. 52

1997/05/29 ltfsstrc.dtx v3.0j
- General: Replaced `\\` by `\MessageBreak`, as suggested by Donald Arseneau. 127

1997/05/29 ltlogos.dtx v1.1f
- `\LaTeXe`: Added `\m@th` so that the LaTeX 2ε logo works with nonzero values of `\mathsurround`. 59

1997/06/16 ltdirchk.dtx v1.0v
- General: documentation improvements mainly from internal/2520 1

1997/06/16 ltfloat.dtx v1.1s
- General: documentation fixes . . . 291

1997/06/16 ltfntcmd.dtx v3.3v
- General: Fix typo in documentation. 191

1997/08/05 ltoutenc.dtx v1.9e
- General: Corrected order of arguments in `\UseTextSymbol` example. 70

1997/08/29 ltoutenc.dtx v1.9f
- General: Added OT4 encoding, provided by Marcin Woliński. . . . 69

1997/09/09 ltdefns.dtx v1.2z
- `\provide@command`: Use `\begingroup` to avoid generating math ords if used in math mode. pr/2573 28

1997/09/15 ltpictur.dtx v1.1g
- `\@getcirc`: Warn if lines become invisible pr/2524 274

- `\@picture@warn`: Macro added pr/2524 . . . 274
- `\@sline`: Warn if lines become invisible pr/2524 . . . 267

1997/10/06 ltcounts.dtx v1.1f
- `\@Roman`: Change `\@Roman` to be fully expandable, so that the result is written properly to files. 105
- `\@slowromancap`: Macro added. . 105

1997/10/08 ltlogos.dtx v1.1h
- `\LaTeX`: Simplify macro (force loading of suitable math fonts once). . . . 59

1997/10/10 ltclass.dtx v1.0y
- `\endfilecontents`: `\@currenvir` in banner . . . 376
  - `\reserved@c` not `\verbatim@out` to save a csname . . . 376
  - Check for text before or after `\end` environment. latex/2636 376
  - Use `\@gobbletwo` . . . 376

1997/10/17 ltfntcmd.dtx v3.3w
- `\check@nocorr@`: Check for vertical mode moved here, from `\DeclareTextFontCommand` (see PR/2646). . . . 194
- `\DeclareTextFontCommand`: Reinstalled `\check@icr` as check is now done in `\check@nocorr@` (see PR/2646). . . . 193

1997/10/20 ltfinal.dtx v1.1a
- `\@uclclist`: Removed `\aa` and `\AA` from `\@uclclist` as these are macros. . . . 383

1997/10/21 ltdefns.dtx v1.2z1
- `\renew@command`: Use `\begingroup`/`\endgroup` rather than braces for grouping, to avoid generating empty math atom. 26

1997/10/21 ltfssbas.dtx v3.0t
- `\define@newfont`: Move `\makeatletter` to `\nfss@catcodes`. . . . 117
- `\nfss@catcodes`: Moved `\makeatletter` from `\try@load@font@shape`. . . . 118

1997/11/09 ltoutput.dtx v1.2c
- `\@specialoutput`: Remove incorrect code: only one `\@emptycol` is needed here . . . 328
- `\@topnewpage`: Documentation of vsize check enhanced . . . 326

1997/11/13 ltfssdcl.dtx v3.0f
- `\DeclareSymbolFont`: (DPC) Really update `\group@list` dont leave new version in `\toks@`. latex/2661 . . . 155
- `\stepcounter`: (DPC) Remove as never used. (Re)defined in ltcounts . . . 151

1997/11/19 ltfloat.dtx v1.1t
- `\@footnotetext`: Missing percent, again . . . 304

1997/11/19 ltoutput.dtx v1.2d
- `\@vtryfc`: Reindent code, to be understandable(DPC). . . . 340

1997/11/20 ltfssdcl.dtx v3.0g
- `\document@select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a csname per math version as no reset list) 152
- `\select@group`: (DPC) inline use of `\stepcounter` (faster, and saves a csname per math version as no reset list) . . . 151

1997/11/23 ltoutenc.dtx v1.9g
- General: Use `\textperthousand`, `\textpertenthousand` and `\textfractionsolidus` not `\textpermill`, `\textpertenmill` and `\textfraction`. /2673 . . . 89

1997/12/17 ltoutenc.dtx v1.9h
- General: Added `\textperthousand` and `\textpertenthousand` 82, 83
  - Added code for textcomp.sty. . 92
  - Added section. . . . 92
  - Added textcomp.sty. . . . 69
  - As in OT1, Added `\leavevmode` at start of `\c`, otherwise the output routine might be invoked within the macro. . . . 83
  - Changed to decimal codes in `\ooalign`. . . . 90
  - Changed to decimal codes. . . . 87
  - Documentation changes and additions. . . . 69
  - Example corrected, braces removed. . . . 69
  - Removed default settings, see next section. . . . 89

1997/12/19 ltoutenc.dtx v1.9i
- General: Documentation corrections. . . . 69

1997/12/20 fontdef.dtx v2.2s
- General: Added documentation . 175

1997/12/31 ltoutenc.dtx v1.9k
- General: Further correction 69

1998/01/12 ltoutenc.dtx v1.9k
- General: Added `\ProvidesPackage` for textcomp.sty . . . 69
  - Adding missing braces and `\ushape`. . . . 90

1998/01/16 ltoutenc.dtx v1.9m
- General: fixed decimal codes. latex/2734 . . . 87

1998/03/04 ltdefns.dtx v1.2z2
- `\@xargdef`: Unnecessary `\expandafter` removed: pr/2758 . 25

1998/03/05 ltoutenc.dtx v1.9n
- General: Added masc/fem ords as in pr/2579 . . . 80

1998/03/20 ltdefns.dtx v1.2z3
- `\@thirdofthree`: Macro added . . 29

1998/03/20 ltoutenc.dtx v1.9o
- General: Added various `\UndeclareTextCommand` declarations for pr/2783 100
  - Documentation added about order of decls 71
  - Documentation added for pr/2783 71
  - Load decls after defaults for speed. 100
- `\UndeclareTextCommand`: Macro added for pr/2783 77

1998/03/21 ltclass.dtx v1.0z
- General: Added to documentation of filecontents 362

1998/03/21 ltclass.dtx v1.1a
- `\@providesfile`: Allow &. Internal/2702 368
- General: Correct to new onlypreamble command list 377

1998/03/25 ltfssbas.dtx v3.0u
- `\showhyphens`: Suppress unnecessary error when used in preamble 123

1998/04/11 fontdef.dtx v2.2t
- General: Added `\mathring` accent (pr2785) 185

1998/04/15 fontdef.dtx v2.2u
- General: Use new syntax for `\DeclareMathDelimiter` 179

1998/04/15 ltfssdcl.dtx v3.0h
- `\@xxDeclareMathDelimiter`: Macro added (pr/2662) 162

1998/04/17 fontdef.dtx v2.2v
- General: Reinsert symbol defs for `<` and `>` chars. 180

1998/04/18 fontdef.dtx v2.2w
- General: Reinsert symbol def for `/` char. 180

1998/05/07 ltclass.dtx v1.1b
- `\@fileswithoptions`: Modify help message for latex/2805 373

1998/05/18 lttab.dtx v1.1j
- `\@endpbox`: Use `\setlength` to set `\hsize`, so that the changes in the calc package apply here. 259
- `\tabular*`: Use `\setlength`, so that calc extensions apply. 251

1998/05/20 ltfinal.dtx v1.1b
- General: Set up lccodes before loading hyphenation files: pr/2639 379
  - Set up uc/lccodes after loading hyphenation files: pr/2639 382

1998/05/28 lterror.dtx v1.2n
- `\@notdefinable`: Added message re ‘end...’ pr/1555 43

1998/06/04 ltboxes.dtx v1.1c
- `\@rule`: Support calc-expressions 239

1998/06/12 ltoutenc.dtx v1.9p
- General: Corrected 130 and 131, see pr/2834 91
  - Renamed `\textmacron` pr/2840 92, 98

1998/06/12 ltoutenc.dtx v1.9q
- `\add@accent`: Explicitly set `\spacefactor` after `\accent` (pr/2877) 75

1998/06/18 lttab.dtx v1.1k
- General: Small addition to documentation 241

1998/07/06 lttab.dtx v1.1l
- General: Small correction to documentation 241

1998/08/17 ltboxes.dtx v1.1e
- General: (RmS) Minor Documentation fixes. 232

1998/08/17 ltclass.dtx v1.1c
- General: (RmS) Minor documentation fixes. 362

1998/08/17 ltdirchk.dtx v1.0w
- General: (RmS) Documentation improvements. 1

1998/08/17 ltfntcmd.dtx v3.3x
- General: (RmS) Minor documentation fixes. 191

1998/08/17 ltfssbas.dtx v3.0v
- General: (RmS) Documentation fixes. 107

1998/08/17 ltfssdcl.dtx v3.0i
- General: (RmS) Corrected minor glitches in changes entries. 149

1998/08/17 ltfssini.dtx v3.0i
- General: (RmS) Minor documentation fixes. 168

1998/08/17 ltlogos.dtx v1.1i
- General: (RmS) Minor documentation fixes. 59

1998/08/17 ltmath.dtx v1.1c
- General: (RmS) Minor documentation fixes. 210

1998/08/17 ltmiscen.dtx v1.1g
- General: (RmS) Minor documentation fixes. 202

1998/08/17 ltspace.dtx v1.2w
- General: Documentation fixes. 48

1998/08/17 preload.dtx v2.1g
- General: (RmS) Minor documentation fixes. 188

1998/09/19 ltoutenc.dtx v1.9r
- `\a`: Added `\string` (pr/2878) 78

1998/11/13 lttab.dtx v1.1m
- `\@array`: Check for hmode to see if something went wrong during parsing (pr/2884) 252

1999/01/05 fontdef.dtx v2.2x
- General: Need special protection for character `>` in `\changes` entry. 173

1999/01/06 ltfssbas.dtx v3.0w
- `\DeclareFontEncoding`: Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) 110

  - `\LastDeclaredEncoding`: Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) 110
- 1999/01/06 ltoutenc.dtx v1.9r
  - `\@strip@args`: New impl for latex/2930 76
  - General: Minor documentation fix. 90
- 1999/01/06 ltoutput.dtx v1.2e
  - `\@makecol`: Added negative vskip, as when processing outputbox below: suggested by Fred Bartlett pr/2892 332
- 1999/01/07 ltdefns.dtx v1.3a
  - `\@ifnextchar`: made long 32
  - `\@newenvb`: made long and brace optional arg. latex/2896 27
  - `\@testopt`: made long and brace optional arg. latex/2896 25
- 1999/01/07 ltdefns.dtx v1.3b
  - `\@ifnextchar`: extra `\long`. latex/2902 32
- 1999/01/07 ltoutenc.dtx v1.9r
  - General: Hackery to allow using fontenc several times 94
    - Hackery to temp support cyrillic uc/lc 93
- 1999/01/13 ltoutenc.dtx v1.9s
  - `\@strip@args`: Simplified solution for latex/2930 76
- 1999/01/18 ltdefns.dtx v1.3c
  - `\@yargd@f`: New implementation DPC /2942 26
- 1999/02/09 ltdefns.dtx v1.3d
  - `\@yargd@f`: catch bad argument forms by re-inserting #3 26
- 1999/02/12 ltfssini.dtx v3.0j
  - `\oldstylenums`: Use `\rmdefault` instead of `cmm` (pr/2954) 170
- 1999/02/24 ltoutenc.dtx v1.9t
  - General: Corrected hackery cyrillic uc/lc list 93
- 1999/03/01 ltdefns.dtx v1.3e
  - `\@ifnextchar`: remove extra `\long`. internal/2967 32
- 1999/04/15 ltpictur.dtx v1.1h
  - `\@getlarrow`: Replaced octal number, CAR 267
  - `\@upvector`: Replaced octal number, CAR 268
  - General: Replaced octal number, CAR 267, 268
    - Replaced octal numbers, CAR 260
- 1999/04/19 ltfloat.dtx v1.1u
  - `\caption`: Made caption an error oustside a float: latex/2815 293
- 1999/04/27 ltboxes.dtx v1.1f
  - `\@parboxto`: (CAR) Changed `\@empty` to `\relax` as flag for natural width: pr/2975 237
- 1999/04/29 ltdefns.dtx v1.3f
  - `\@yargd@f`: Full expansion and conversion needed for digit in new version, see pr/3013 26
    - New macro added 26
- 1999/06/10 ltoutenc.dtx v1.9u
  - General: Ensure that we also forget old options (pr/2888) 94
- 1999/06/12 ltoutenc.dtx v1.9v
  - General: Extend `\@uclclist` only once 94
- 1999/10/09 ltmath.dtx v1.1e
  - `\active@math@prime`: Macro added, see PR 3104. 213
  - `\prime@s`: Introduce `\active@math@prime`. 213
- 1999/10/09 ltoutput.dtx 1.2f
  - `\@activechar@info`: Reset definition of active prime character (used in math mode) 335
- 1999/10/28 ltoutenc.dtx v1.9w
  - `\add@accent`: Give `\accent@spacefactor` a default definition (pr/3084) 75
- 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. 93
- 2000/01/07 ltmiscen.dtx v1.1h
  - `\@verbatim`: Disable hyphenation even if the font allows it. 207
- 2000/01/15 ltpictur.dtx v1.1i
  - `\@upvector`: Removed space at end-of-line, CAR 268
- 2000/01/30 ltfntcmd.dtx v3.3y
  - `\DeclareTextFontCommand`: Use `\hmode@bgroup` now (pr/3160) 193
- 2000/01/30 ltoutenc.dtx v1.9y
  - General: Use `\hmode@bgroup` where applicable (pr/3160) 81–83, 86, 88–90
  - `\add@accent`: Use `\hmode@bgroup` where applicable (pr/3160) 75
  - `\hmode@bgroup`: Macro added 75
- 2000/01/30 ltoutenc.dtx v1.9z
  - `\@use@text@encoding`: Macro reimplemented (pr/3160) 76, 77
  - `\add@accent`: Macro reimplemented (pr/3160) 75
  - `\hmode@start@before@group`: Macro added (pr/3160) 77
- 2000/05/19 ltmiscen.dtx v1.1i
  - `\enddocument`: Reset `\AtEndDocument` for latex/3060 203
- 2000/05/26 ltpage.dtx v1.0j
  - `\@markright`: Reimplementation to fix expansion error (pr/3203). 312
  - `\leftmark`: Use `\@empty` instead of brace group (pr/3203). 312

  - `\markright`: Reimplementation to fix expansion error (pr/3203). 312
  - `\rightmark`: Use `\@empty` instead of brace group (pr/3203). 312
- 2000/06/02 ltpage.dtx v1.0k
  - `\@markright`: Small adjustment to give slightly less expansion, CAR 312
  - `\markright`: Small adjustment to give slightly less expansion, CAR 312
    - Tidied 1.0j reimplementation, CAR 312
- 2000/07/11 ltmiscen.dtx v1.1j
  - `\enddocument`: Fix typo in warning 203
- 2000/07/12 ltoutput.dtx 1.2g
  - General: Ensure that rule is in `\normalcolor` 358
- 2000/07/12 ltoutput.dtx 1.2i
  - `\@makecol`: Removed negative vskip, as it gives unacceptable results when the depth is large: pr/3189 332
- 2000/07/19 ltoutput.dtx v1.2h
  - `\@writesetup`: Reset and restore `\@if@newlist` for internal/3231 335
- 2000/08/30 ltoutenc.dtx v1.91
  - `\@use@text@encoding`: Rearranged but no change to final code, CAR (pr/3160) 76
  - `\add@accent`: Rearranged but no change to final code, CAR (pr/3160) 74
- 2000/09/01 ltfinal.dtx v1.1d
  - `\errhelp`: Set error help empty at very end (pr/449 done correctly). 385
- 2001/01/07 ltoutput.dtx v1.2j
  - `\@writesetup`: And do it in the right macro (pr/3286) 335
- 2001/02/16 ltxref.dtx v1.1k
  - `\@newl@bel`: Added an extra grouplevel (PR3250), jlb 200
- 2001/05/25 ltclass.dtx v1.1d
  - `\@providesfile`: Explicitly set catcode of `\endlinechar` to 10 (pr/3334) 368
- 2001/05/25 ltdirchk.dtx v1.0x
  - General: Explicitly set catcode of `\endlinechar` to 10 (pr/3334) 3
- 2001/05/28 ltoutenc.dtx v1.93
  - General: Added composites for compatibility with T1, pr/3295 82
    - Changed the effect of `\.\i`, pr/3295 84
- 2001/06/02 fontdef.dtx v2.2y
  - General: Provide default cfg files (pr/3264) 187
- 2001/06/04 fontdef.dtx v2.2z
  - General: Guard against math active equal and pipe sign in `\models` (pr/3333) 184
    - Guard against math active equal sign in `\Relbar` (pr/3333) 184
- 2001/06/04 ltclass.dtx v1.1e
  - `\@providesfile`: But only if it is a char (pr/3334) 368
- 2001/06/04 ltdirchk.dtx v1.0y
  - General: But only if it is a char (pr/3334) 3
- 2001/06/04 ltpictur.dtx v1.1j
  - `\@sline`: Don't warn for exactly zero pr/3318 267
- 2001/06/04 ltvers.dtx v1.0i
  - General: Check for old format disabled 21
- 2001/06/05 ltoutenc.dtx v1.94
  - General: Text composite Commands need kludges for ',' – see tlb1903.lvt 82
- 2001/08/26 ltclass.dtx v1.1f
  - `\@providesfile`: Readded setting of space char (pr/3353) 368
- 2002/02/24 ltplain.dtx v1.1x
  - `\loggingall`: Macro added 20
  - `\loggingoutput`: Macro added 20
  - `\showoutput`: Use newly added `\loggingoutput` 20
  - `\tracingall`: Use newly added `\loggingoutput` 20
- 2002/06/16 ltoutenc.dtx v1.95
  - General: Added `\textbardbl` (pr/3400) 86
    - Added default for `\textbardbl` (pr/3400) 79
- 2002/06/17 ltoutenc.dtx v1.95
  - General: Corrected `\c` for T1 (pr/3442) 83
    - Definition of `\textexclamdown` changed (pr/3368) 81
    - Definition of `\textquestiondown` changed (pr/3368) 81
- 2002/06/18 ltoutenc.dtx v1.95
  - General: Changed def for `\textregistered` to avoid small caps (pr/3420) 80
- 2002/10/01 ltfloat.dtx v1.1v
  - `\thempfootnote`: Use braces around `\itshape` to keep font change local (pr/3460). 303
- 2002/10/02 ltfssbas.dtx v3.0x
  - `\DeclareFontSubstitution`: Adding `\LastDeclaredEncoding` introduced a bug as on some occasions that macro name was stored in the internal lists instead of the actual encoding. (pr/3459) 110
- 2002/10/28 ltlists.dtx v1.0s
  - `\endtrivlist`: Check for math mode (pr/3437) 227

2002/10/28 ltoutenc.dtx v1.96
- General: coding change, to follow bug fix by DEK in plain.tex (pr/3469) 82, 88

2002/12/13 ltbibl.dtx v1.1n
- `\@citex`: Added `\leavevmode` in case citation is at start of paragraph (pr/3486) 308

2003/01/01 ltfntcmd.dtx v3.3z
- General: Code checked and documentation extended by Chris 193

2003/05/18 ltbibl.dtx v1.1o
- `\nocite`: Check if we are after `\document` 309

2003/08/27 ltpictur.dtx v1.1k
- `\@bezier`: added missing displacement pr/3566 278
- `\@sline`: check for `\@linechar` being empty pr/3570 266

2003/10/13 ltfinal.dtx v1.1e
- General: Added extra `\lccode` for `\-` and `\textcompwordmark` . 380

2003/12/16 ltoutput.dtx v1.2k
- `\@makecol`: Ensure that `\@elt` has a defined state (pr/3586) . . . 332

2003/12/30 ltpictur.dtx v1.1j
- `\@getcirc`: issue warning if circle size can't be met pr/3473 . . 274

2004/01/03 ltoutenc.dtx v1.99b
- General: Added `\textogonekcentered` (pr/3532) 83
  - Added composites for `\k` (pr/3532) 86
  - Use `\ooalign` for `\k` (pr/3532) 83

2004/01/04 ltbibl.dtx v1.1p
- `\nocite`: Changed error message 309

2004/01/04 ltoutenc.dtx v1.99c
- General: More adjustments for ogonek (pr/3532) 83

2004/01/23 ltdefns.dtx v1.1g
- `\@newenva`: Use kernel version of `\@ifnextchar` (pr/3501) 27
- `\@testopt`: Use kernel version of `\@ifnextchar` (pr/3501) 25
- `\@xargdef`: Use kernel version of `\@ifnextchar` (pr/3501) 25
- `\@xdblarg`: Use kernel version of `\@ifnextchar` (pr/3501) 33

2004/01/23 ltdefns.dtx v1.3g
- `\kernel@ifnextchar`: Added macro (pr/3501) 32

2004/01/28 ltclass.dtx v1.1g
- `\@providesfile`: Use kernel version of `\@ifnextchar` (pr/3501) . 368

2004/01/28 ltvers.dtx v1.0k
- General: Check for old format made 5 years (pr/3601) 21

2004/02/02 fontdef.dtx v2.3
- General: Many things from here on made robust 183

2004/02/04 fontdef.dtx v2.3a
- General: Added bigtriangle synonyms for stmaryrd 181

2004/02/04 ltspace.dtx v1.3
- `\nobreakdashes`: (Macro added . 56

2004/02/06 ltoutenc.dtx v1.99d
- `\@inmathwarn`: New command added to fix severe bug: pr/3563 72

2004/02/07 ltoutput.dtx v1.2l
- `\@doclearpage`: Empty kludgeins box if necessary, pr/3528 . . . 331

2004/02/13 ltoutenc.dtx v1.99e
- General: Documentation fixes: typos 69

2004/02/15 ltbibl.dtx v1.1q
- `\@cite@ofmt`: Added hook with default value `\hbox` 310
- `\@citex`: Changed to use a hook with default value `\hbox` . . . 308

2004/02/15 ltspace.dtx v1.3a
- `\nobreakdashes`: (Added spacefactor setting 56

2004/10/20 ltoutput.dtx v1.2m
- `\@makecol`: Removed dead code . 332

2005/07/27 ltfssdcl.dtx v3.0j
- `\DeclareMathAlphabet`: (MH) Make document commands robust 157
- `\DeclareSymbolFontAlphabet`: (MH) Make document commands robust 165
- `\new@mathalphabet`: (MH) Make document commands robust 157
- `\non@alpherr`: (MH) Change because command is now properly robust 152
- `\SetMathAlphabet`: (MH) Make document commands robust 158

2005/09/27 ltoutenc.dtx v1.99g
- General: Replace `\sh@ft` by `\ltx@sh@ft` 81, 83, 88

2005/09/27 ltplain.dtx v1.1y
- `\ltx@sh@ft`: New macro 19
- `\sh@ft`: Macro no longer used but left for compatibility 19

2005/11/08 ltoutenc.dtx v1.99h
- General: Added `\ij` and `\IJ` from babel. (pr/3771) 79, 82, 83

2005/11/10 ltmath.dtx v1.1g
- `\[`: (MH) Fixed potential problem in `\[` (pr/3399). 214
- General: (MH) Minor documentation fixes. 210

2006/05/18 ltboxes.dtx v1.1g
- `\@parboxto`: Ensure `\@parboxto` holds the value of `\@tempdimb` not the register itself (pr/3867) 237

2007/08/05 ltclass.dtx v1.1h
- `\@fileswithoptions`: Prevent loss of brackets PR/3965 372

2007/08/06 ltcntrl.dtx v1.0h
    `\@fornoop`: Really make defs long 37
2007/08/31 ltfssdcl.dtx v3.0l
    `\SetSymbolFont@`: Font warning changed to info for encoding change (pr/3975) 156
2009/09/24 ltvers.dtx v1.0l
    General: Stop checking for old format 21
2009/10/20 ltfssdcl.dtx v3.0m
    `\in@`: More robust thanks to Heiko. 149
2009/10/28 ltoutenc.dtx v1.99k
    General: Added Latin Modern and TeX Gyre subsets 101
2009/11/04 ltoutenc.dtx v1.99l
    General: Added more Latin Modern and TeX Gyre subsets 101
2009/12/14 ltfntcmd.dtx v3.4a
    `\ifmaybe@ic`: Macro added 195
    `\maybe@ic@`: Use switch `\ifmaybe@ic` instead of `\if@tempswa` 195
    `\t@st@ic`: Use switch `\ifmaybe@ic` instead of `\if@tempswa` 195
2010/08/17 ltmiscen.dtx v1.1k
    `\enddocument`: Use braces around `\input` arg (pr/4124) 203
2010/08/17 ltmiscen.dtx v1.1l
    `\enddocument`: Change of plan: use `\@@input` instead (pr/4124) . 203
2011/05/08 ltfssdcl.dtx v3.0n
    `\in@`: Simplified thanks to Bruno. 149

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

\! b152, b154, z144, N114
\" l181, l280, l318, l356, l364, l436, l468, l495, l503, l509, l513, l519, l523, l529, l535, l542, l543, l549, l553, l593, l636, o367, N115
\# a9, a22, b6, b14, b219, d278, o306, o354, N98
\$ a21, b4, b13, d277, l261, l343, l350, l425, l648, l655, N99
\% a22, a52, a54, a74, b14, b217, d278, l386, l388, o356, L492, L493, N100
\& a21, b5, b13, b218, d277, L111, N101
\' b239, l182, l281, l320, l354, l361, l438, l448, l454, l456, l459, l461, l469, l475, l481, l483, l486, l488, l496, l500, l507, l511, l516, l521, l524, l526, l533, l538, l539, l546, l551, l554, l594, l638, l657, l659, l660, l661, l664, l666, l667, l668, l670, l671, o366, s154, t172, y145, z151, B181, C63, K466, N116
\(z168, z193
\) b239, z168, z194
* o359, z148, L426, L494
\+ C63
\, b153, b155, i187, t414, y145, z7, z8, z40, z108, z110, z113, z127, z144
\- b121, d9, d11, i178, l315, l316, l431, l632, l633, o361, y145, B180, C63, N60
\. b152, b154, k39, l183, l282, l351, l352, l370, l444, l445, l471, l472, l498, l595, l662, l669, o360
\/ a44, d12, o274, o362, L110
\: b153, b155, d270, d271, z149
\; b153, b155, t408, z128, z144
\< l432, l586, o357, y145, C62, C100
\= l184, l283, l369, l596, s154, B181, C62
\> l429, l587, o358, y145, z144, z149, C62
\? b152, b154, N116
\@ a12, d272, d273, g22, i190, j2, L24, L32, N108
\@@ a254, a255, f15, f19, f20, f21, f22, f24, f27, f28, f30, f31, k208, k224, p494, p498, p499, C208, C209, C210, C220, K10, K11
\@@defaultsubs o414
\@@enc@update l138, o207, o211
\@@end a16, a169, d8, k186, k187, s121, y39, y49, M18, N198, N219
\@@endpbox C175, C206, C354
\@@eqncr z228, z246, z249, z329
\@@fileswith@pti@ns L184, L354
\@@hyph d9
\@@hyphenation l160
\@@if@newlist K476, K523
\@@ifdefinable d101, l17
\@@input a15, d7, d293, k162, k163, k172, o307, y19
\@@italiccorr d12, v96, v100
\@@line B289
\@@math@bgroup v114, v121
\@@math@egroup v111, v111
\@@par d6, h4, *46*, y49, y104, y108, y111, A82, A85, B162, B179, C181, F50, F101, K205
\@@patterns l160
\@@protect d228, d234, d243
\@@startpbox C175, C206, C354
\@@underline B259, B262, B263
\@@unprocessedoptions L336, L398
\@@warning g188
\@Alph m38, m54
\@DeclareMathDelimiter r604, r623
\@DeclareMathSizes o181, o182, o184
\@Esphack i77, G126, G137
\@M b21, b196, b197, d23, d25, i27, i28, i29, i30, i31, i32, i33, i34, i57, p401, p424, z238, A184, C58, F50, F83, F101, F113, F154, K145, K146, K206
\@MM b21, G275, K249
\@Mi e3, K117
\@Mii e3, G49, G119, G137, G190, K245, K898
\@Miii e3, G51, G192, K248
\@Miv e6, G120, G126, K220
\@Roman m36, m42
\@TeXversion *1*, *5*, a249, g31

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@acci . . . s154, B181
\@accii . . . s154, B181
\@acciii . . . s154, B181
\@acol . . . C150, C160, C230, C231, C243, C244, C247, C264, C277, C285, C295
\@acolampacol . . . C228, C245, C247, C254, C262, C294, C297
\@activechar@info . . . K457
\@addamp . . . C221, C230, C231, C246, C260, C295, C296
\@addfield . . . C45, C55, C77, C84, C116, C127, C129
\@addmarginpar . . . K283, K1149
\@addtobot . . K717, K804, K868, K920
\@addtocurcol . . . K280, K808
\@addtodblcol . . . K649, K1032
\@addtofilelist a48, a50, k54, k162, k205, s111, s129, s133, s140, s143, s150, s153, N67, N70, N239
\@addtonextcol . . . K648, K955
\@addtopreamble . . . C279, C292, C298, C299, C300, C302, C314
\@addtoreset . . . m16, m28, m33
\@addtotoporbot . . K754, K914, K1001
\@afterheading . . . F75, F108
\@afterindentfalse . . . F28
\@afterindenttrue . . F26, F107, F153
\@alph . . . m37, m50, G254
\@ampacol . . . C228, C245, C256, C297
\@arabic . . . m32, m34, m40, G252
\@argarraycr . . . C185, C186
\@argdef . . . d47
\@argrsbox . . . B269
\@argtabularcr . . . C192, C193
\@array . . . C163, C164
\@arrayacol . . . C150, C228
\@arrayclassiv . . . C151, C299
\@arrayclassz . . . C150, C245
\@arraycr . . . C152, C183, C185
\@arrayparboxrestore B176, B190, C352
\@arrayrule . . . C277, C279, C283, C285, C287, C314
\@arstrut . . . C174, C207, C311
\@arstrutbox . C167, C200, C311, C353
\@author . . . F5
\@autoerr . . . b75, b247, b255, g19, g88, g119, g150, g213, g220, g223, g226, g230, g236, g239, g242, g246, g249, g258, g264, g267, g270, g273, g278, g281, g286, k174, l56, o483, r87
\@autoload . . . d286, g213, p417, C146, D15, K1212
\@auxout . . . I7, I8, I19, I29, I37, I43, k81, k87, k103, k118, x33, F145
\@backslashchar d191, g216, g218, t185, L459
\@badcrerr . . . g274
\@badend . . . g233, y65
\@badlinearg . . . g260, D79, D88, D89, D93, D137, D142, D153
\@badmath . . . g237, z168, z169, z172, z184, z189, z277, z291
\@badpoptabs . . . g243, C76, C138
\@badrequireerror . . . L124, L406
\@badtab . . . g247, C24, C78, C99, C105, C112, C135
\@begin@tempboxa . . . B11, B26, B114, B162, B270, B278
\@begindocumenthook I33, k48, k51, o301, L365, L379
\@begindvi . . . K502, K528
\@begindvibox . . . K67, K529
\@beginparpenalty i30, z280, z294, A23, A160
\@begintheorem . . . E30, E35
\@bezier . . . D342, D343
\@bibitem . . . I3, I8
\@biblabel . . . I4, I54
\@bitor . . . K15, K672, K702, K769, K851, K860, K989, K1094
\@botlist . . . K46, K308, K310, K535, K556, K565, K566, K731, K734, K769, K771, K860, K862
\@botnum . . . G172, K90, K728, K729, K734, K738, K745, K1024, K1029, K1257, K1279
\@botroom G173, K91, K731, K734, K1258
\@boxfpsbit . . . K1327, K1329, K1334
\@break@tfor . . . f31, k157, v81
\@bsphack . . . I39, i9, i63, i138, i154, x32, G48, G189, H6, H18, H23, H35, K1229
\@caption . . . G12, G14
\@captype G5, G9, G12, G36, G84, K1291
\@car . . . *24*, d30, j14, l82
\@carcube . . . d32, d104
\@cclv . . . b16, K250, K254, K306, K307, K335, K359, K363, K364
\@cclvi . . . b21, b50, b53, b56
\@cdr . . . *24*, d30, d251, d252
\@centercr . . . y68, y76, y83, y89

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@centering z208, z209, z216, z219, z222, z322, z326
\@cflb K532
\@cflt K532
\@changed@cmd . l3, l68, l178, o106, o215
\@changed@x l3, l166, l174
\@changed@x@mouth l166, l174
\@charlb k121, k129
\@charrb k123, k129
\@chclass C241, C242, C303, C316, C321
\@check@c d162, d164
\@check@eq d168, d169, d173
\@checkend y11, y61, y64
\@chnum C249, C268, C303, C318, C319, C320
\@circ D308, D309, D310
\@circle D296, D297
\@circlefnt D47, D49, D257, D275, D301, D316, D333
\@cite I16, I52
\@cite@ofmt I24, I53
\@citea I15, I17
\@citeb I16, I18, I19, I20, I23, I24, I41, I42, I43, I44, I45
\@citex I13, I14
\@classi C241, C275
\@classii C241, C289
\@classiii C241, C294
\@classiv C151, C162, C242
\@classoptionslist L9, L153, L164, L281, L282, L517
\@classv C242, C300
\@classz C150, C161, C241
\@cline C335
\@clnht . . . D95, D96, D104, D106, D108, D118, D125, D151, D327
\@clnwd D97, D103, D107, D109, D110, D327
\@cls@pkg L87, L88, L316, L345, L382, L391, L393, L410
\@clsextension L16, L41, L52, L70, L94, L120, L137, L153, L163, L203, L218, L226, L280, L349, L357, L383
\@clubpenalty k9, k19, A186, F89, F118
\@colht . . . k16, G171, G173, G176, G179, G180, K95, K179, K190, K199, K200, K311, K323, K350, K383, K417, K423, K427, K437, K442, K524, K595, K612, K654, K674, K707, K1108, K1392, N12
\@colnum G174, K92, K737, K782, K849, K850, K877, K885, K927, K987, K988, K1024, K1029, K1253, K1272, K1446
\@colroom k17, K96, K200, K221, K222, K233, K236, K311, K595, K736, K781, K845, K848, K876, K983, K986, K1254, K1402, K1406, K1451, N11
\@combinedblfloats K568, K1492
\@combinefloats K377, K532
\@comdblflelt K568
\@comflelt K538, K554, K568
\@cons *24*, d29, m33, G118, G135, G234, K185, K678, K692, K711, K713, K880, K948, K1017, K1110, K1133, K1150, K1151, K1452
\@contfield C52, C128, C140
\@ctrerr g228, m53, m57, m60
\@curfield C16, C43, C49, C53, C54, C56, C121, C122
\@curline C16, C29, C41, C46, C55, C56, C57, C81, C82, C94, C119, C120
\@curr@enc l119, l121
\@currbox G56, G87, G91, G118, G135, G155, G157, G159, G198, G201, G206, G210, K161, K162, K173, K174, K176, K177, K185, K261, K262, K648, K649, K842, K844, K874, K878, K880, K895, K936, K948, K977, K1006, K1017, K1048, K1052, K1063, K1069, K1071, K1075, K1080, K1089, K1103, K1110, K1133, K1151, K1160, K1297, K1298, K1327, K1357, K1362, K1407, K1410, K1422, K1430, K1447, K1452
\@currdir *1*, *4*, a55, a77, a79, a85, a87, a93, a95, a100, a102, a112, a125, a190, a203, a216, L436
\@current@cmd l25, o219
\@currentlabel x34, x37, x40, z212, z307, B243, G277
\@currenvir g234, y3, y55, y65, A112, B74, L453, L459, L467, L471, L477
\@currenvline g234, y56, y66, B75
\@currext L15, L23, L31, L93, L94, L137, L146, L153, L163, L213, L222, L307, L308, L313, L314,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

L319, L325, L329, L331, L333, L335, L337, L338, L341, L347, L349, L357, L375, L383, L399, L400
\@currlist G118, G234, K48, K261, K312, K315, K1150
\@currname k216, k217, L14, L22, L30, L85, L87, L93, L146, L222, L306, L308, L331, L333, L335, L337, L338, L375, L391, L393, L400, L410
\@currnamestack L20
\@curroptions L146, L154, L176, L400, L401
\@currsize s58
\@currtype K100, K669, K670, K671, K672, K769, K851, K860, K989, K1094, K1297, K1299, K1300, K1303
\@curtab C11, C28, C77, C78, C79, C85, C86, C89, C93, C94, C98, C133, C134
\@curtabmar C11, C27, C28, C40, C46, C80, C93, C97, C98
\@d@r a108, a109
\@dashbox D196, D197, D198, D199, D200, D203, D206, D208, D217, D219, D220, D221, D222, D225, D228, D231, D329
\@dashcnt D190, D191, D192, D193, D194, D195, D205, D207, D210, D211, D212, D213, D215, D216, D227, D230, D329
\@dashdim D189, D190, D191, D192, D194, D197, D199, D200, D201, D205, D207, D209, D210, D211, D212, D215, D219, D221, D222, D223, D229, D232, D329
\@date F7
\@dbflt G32, G162
\@dblarg *23*, d275, F37, F125, G12
\@dbldeferlist . G135, K51, K321, K326, K328, K613, K620, K621, K1094, K1096, K1133, K1135
\@dblfloat G31
\@dblfloatplacement k25, G177, K325, K1251, K1498
\@dblflset G26
\@dblfpbot G185, K1541
\@dblfpsep G184, K1541
\@dblfptop G183, K1541
\@dbltoplist K50, K180, K183, K185, K321, K322, K573, K577, K579, K580, K1105, K1110
\@dbltopnum G178, K88, K108, K186, K188, K584, K1045, K1046, K1109, K1112, K1120, K1140, K1145, K1261
\@dbltoproom G179, G181, K89, K1048, K1051, K1052, K1061, K1062, K1065, K1068, K1071, K1074, K1079, K1083, K1088, K1107, K1262
\@dec@text@cmd l3
\@declaredoptions L8, L127, L150, L166, L181, L363
\@declareoption . . . L125, L126, L134
\@defaultsubs o404, o414, y26
\@defaultunits o185, o245, p133, p135
\@defdefault@ds . . . L125, L130, L135
\@deferlist K49, K308, K317, K318, K596, K604, K605, K851, K853, K948, K950, K989, K991, K1017, K1019
\@definecounter m12, m25, z197, A217, A218, A219, A220, E8, E16, G251, G253
\@depth d13, p145, t464, t465, t467, t468, B258, B288, C169, C201, D127, D178, D181, D196, D203, D376, K1189
\@dir a107, a110, a112, a114, a115
\@dischyph d11, B180
\@doclearpage K246, K297
\@documentclasshook L3, L285
\@doendpe y62, A123
\@dofilelist k214, k230, y21
\@donoparitem A134, A148
\@dot D296, D309
\@dotsep F160
\@dottedtocline F149
\@downline D175, D179, D184
\@downvector D146, D184
\@eha l47, g195, g219, g222, g225, g235, g238, g280, k88, l54, l939, l949, o33, o77, o119, o162, o201, o256, p106, r25, r66, r127, r242, r274, r315, r360, r365, r420, r528, r532, r536, r570, r580, r664, r669, r672, r704, r707, r761, r764, r767, r834, r840, v129, y54, K1223, K1239
\@ehb g195, g229, g263, g266, g269, K182, K314

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\@ehc d97, d124, g195, g274, g277, g285, g288, y130, y141, z253, A210, F4
\@ehd g195, g232, g241, g245, g248, g256, r85, C91, C100, G6, L251
\@elt d29, k122, m20, m24, K8, K11, K15, K41, K42, K43, K44, K374, K538, K549, K554, K564, K576, K578, K606, K622, K639, K652, K659, K684, K687, K696
\@empty f14
\@emptycol K146, K193, K196, K225, K229
\@end@tempboxa B20, B29, B119, B175, B276, B286
\@enddocumenthook y10, L365, L380
\@endfloatbox G115, G132, G142
\@endparenv A120, A123
\@endparpenalty i31, z281, z295, A23, A124
\@endpbox C175, C206, C236, C301, C352, C355
\@endpefalse y59, A126, A127, A128, B77
\@endpeltrue A128
\@endpetrue A124, A125
\@endtheorem E13, E19, E25, E35
\@enlargepage K1203, K1208, K1214
\@ensuredmath z264, z266
\@enumctr A224, A227, A228
\@enumdepth A216, A222, A223, A224
\@eqcnt z205, z250, z255, z309, z324, z325, z327
\@eqncr z217, z235, z256, z257, z311
\@eqnnum z199, z200, z254, z268, z302
\@eqnsel z205, z323
\@eqnswfalse z234
\@eqnswtrue z207, z213, z255, z308
\@eqpen z205, z238, z240, z247
\@err@ g40, g44, g47, g55, g67, g71, g74, g82
\@esphack I50, i11, i69, i143, i160, x35, G240, H17, H19, H34, K1231
\@evenfoot J12, J15, K492
\@evenhead J12, J15, K491
\@expandtwoargs d189, L74, L152, L166, L190
\@expast C209, C237
\@failedlist K637, K660, K672, K678, K692, K702
\@fcolmadefalse K628
\@fcolmadetrue K690
\@filef@und k144, k154, k162, k172
\@filelist k53, k204, k205, k216, s111, s129, s140, s150, N67, N223, N239
\@fileswfalse k64
\@fileswith@pti@ns L124, L184, L276, L277, L279, L303, L354
\@fileswith@ptions L271, L272, L274, L278
\@fileswithoptions L203, L210, L218, L269
\@fileswtrue k7
\@finalstrut B247, B287, C353, G282
\@firstampfalse C224, C247, C264
\@firstamptrue C232
\@firstcolumnfalse K1471
\@firstcolumntrue k22, K79, K155, K1477
\@firstofone I18, I42, d184, k47, l73, l118, p300, r49, r108, r615, y9, z262, C340, G10
\@firstoftwo a34, d184, d247, d274, k155, l102, l911, l927, r619, x19, J16, L48, L64, L77
\@firsttab C2, C65, C66, C67, C97, C109
\@flcheckspace K731, K767, K1398
\@flfail K660, K685, K702, K711
\@float G26, G32
\@floatboxreset G97, G99
\@floatpenalty G3, G49, G51, G54, G116, G119, G124, G126, G133, G137, G190, G192, G196, G200, G234
\@floatplacement k25, G169, K130, K157, K201, K353, K1252
\@flsetnum K728, K764, K849, K987, K1045, K1366
\@flsettextmin K825, K973, K1382
\@flstop K1268
\@flsucceed K653, K661, K684, K713
\@fltovf g268, G89, G201
\@flupdates K734, K779, K1443
\@flushglue e17, y77, y83, y90, y103, A76, B187
\@fnsymbol m39, m58
\@font@info o108, o146, o152, o285, o334, o452, p30, p38, p46, p74, p87, p126, p154, p168, p179, p193, p209, p215, p228, p235, p242, p247, p257, p269, p281, p465, p481, p490, p503, p532, p544, r135, r150, r184,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

r225, r294, r300, r344, r357, r440, r519, r561, r654, r802, r831
\@font@warning o4, o400, o405, p19, p33, p41, p49, p61, p77, p448, p464, p480, p489, p502, p531, p543, q20, y23, N73
\@fontswitch v109, v111
\@footnotemark G264, G270, G288, G294, G295
\@footnotetext B217, G264, G270, G271, G304, G310
\@for I16, I41, f16, k99, k216, L78, L150, L164, L176, L181, L196, L401
\@forloop f19, f20
\@fornoop f15, f23, f29
\@fortmp f17, f18, f26
\@fpbot G185, K658, K1535
\@fpmin G176, G182, K94, K689, K1259, K1458
\@fps G37, G38, G40, G43, G60, K1289, K1291, K1294
\@fpsadddefault G41, G44, K1286
\@fpsep G184, K656, K665, K706, K1535
\@fpstype K725, K746, K747, K761, K792, K793, K815, K817, K820, K822, K872, K928, K929, K963, K965, K968, K970, K1042, K1057, K1059, K1077, K1086, K1121, K1122, K1282, K1298, K1300, K1302, K1305, K1306, K1307, K1309, K1310, K1314, K1315, K1317, K1318, K1352, K1354, K1356, K1368, K1370, K1384, K1386, K1415, K1418, K1429
\@fptop G183, K655, K1535
\@frameb@x B104, B118, B120
\@framebox B107, B108
\@framepicbox B107, B141
\@freelist G56, G198, G199, K41, K161, K375, K550, K565, K579, K661, K1150, K1151
\@getcirc D247, D271, D299
\@getfpsbit K722, K758, K1039, K1325
\@getlarrow D144, D152, D154
\@getlinechar D90, D129
\@getpen i7, i10, i21, i55
\@getrarrow D145, D152, D161
\@glossaryfile H21, H22, H31
\@gnewline i46, i48, i49
\@gobble I11, I25, I26, d79, d102, d181, d191, d210, d214, d292, f6, f9, g109, g140, g171, g180, i42, i207, k54, k204, l29, l889, o401, p299, q16, r28, r30, r188, r199, r258, r305, r306, r335, r341, r349, r354, r372, r386, r396, r405, r418, r435, r444, r522, r564, r657, r720, r793, r824, s133, s143, s153, F126, F127, F128, F129, F130, F146, G7, K498, K499, K500, K696, K1247, K1459, L239, L422, L446, L451, N70, N168, N174, N239
\@gobblecr i205, i206
\@gobblefour d181, r24, r185, r296, r298, r302, r304, r314, r318, r442, r494, L453
\@gobbletwo d148, d149, d181, f12, k26, o406, r102, y16, y24, J11, J13, L445, N79
\@gtempa d95, d96, d154, d156, k183, k184, k186, k187, k188, C3, C5, C6, C7, C8, L84, L85, L95, L97
\@halfwidth D2, D48, D50, D57, D127, D177, D180, D196, D203, D217, D227, D230, D335, D361, D374, D375, D376
\@halignto C152, C156, C159, C173
\@hangfrom F49, F100, F121
\@height b190, d13, i148, i156, l247, l249, p144, t246, t464, t465, t467, t468, B88, B93, B127, B137, B258, B288, C168, C201, C327, C344, D127, D178, D181, D196, D203, D219, D226, D293, D375, K1189
\@highpenalty i56, N3
\@hightab C11, C23, C25, C65, C77, C86, C87, C102, C133, C134
\@hline D81, D126, D143
\@holdpg K103, K250, K252, K253, K258, K259, K260
\@hspace i191, i192
\@hspacer i191, i193
\@hvector D139, D143
\@icentercr y71, y72
\@iden d187
\@if d143, d144, d146
\@if@pti@ns L74, L76, L82
\@if@ptions L69, L70, L73, L75, L314
\@ifatmargin C57, C97
\@ifclasslater *364*, L51
\@ifclassloaded *364*, L40

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\@ifclasswith *364*, L69
\@ifdefinable *23*, d51, d53, d99, d101, d212, l14, l17, m11, n3, s54, B54, E7, E15, E22
\@iffileonpath k140, k148
\@ifl@aded L40, L41, L44, L50, L313
\@ifl@t@r L56, L59, L66, L260
\@ifl@ter l875, l876, L51, L52, L55, L58, L341
\@ifl@ter@@ l875, l876
\@ifnch d257, d259, d271
\@ifnextchar I3, I13, *23*, a45, d253, d258, d274, i44, i206, k163, m13, p371, y70, z203, A133, B5, B7, B10, B31, B56, B57, B61, B65, B106, B107, B109, B142, B146, B150, B154, B195, B199, B203, B250, B266, B268, C59, C163, C185, C192, D12, D63, D74, D263, E3, E5, E28, G27, G162, G203, G262, G285, G302, K157, K1270, L90, L255, L270, L275
\@iforloop f21, f22
\@ifpackagelater *364*, L51
\@ifpackageloaded *364*, L40
\@ifpackagewith *364*, L69
\@iframebox B110, B111, B112
\@iframepicbox B142, B143
\@ifstar *23*, d40, d274, i38, i132, i191, o181, q111, y69, y136, z237, C58, C184, C191, D73, D296, F35, F125, K1198, L125, L147
\@ifundefined I20, I44, *23*, d96, d103, d123, d131, d156, d167, d245, l891, m3, m7, m16, o75, o161, p384, r220, x23, y44, y53, E21, J3, J7, L38, L115, L177, L483, L486
\@ignorefalse y4, y58, y63, G239
\@ignoretrue i82, y4, y7, z196, z199, z231, z332
\@iiiminipage B197, B201, B204, B205, B206
\@iiiparbox B148, B152, B155, B156, B157, B234
\@iiminipage B200, B202
\@iinput k163, k164
\@iiparbox B151, B153
\@iirsbox B268, B277
\@imakebox B10, B25, B63
\@imakepicbox B31, B32, B68, B144
\@iminipage B196, B198
\@include k89, k90
\@index H18, H19, H35
\@indexfile H4, H5, H14
\@inlabelfalse A28, A104, A174, K140
\@inlabeltrue A28, A168
\@inmatherr g282, A112, A132, D296
\@inmathwarn l3
\@input k28, k93, k171, F135
\@input@ I31, k108, k173, o344
\@inputcheck a17, a138, a139, a142, a150, d24, k3, k135, k136, k143, k152, k153, k156, L433, L434, L441
\@insertfalse K813, K961, K1037
\@inserttrue K741, K786, K900, K1115
\@invalidchar g288
\@iparbox B147, B149
\@irsbox B266, B268, B269
\@isavebox B61, B62
\@isavepicbox B66, B67
\@ishortstack D64, D72
\@istackcr D74, D75
\@itabcr C59, C60
\@item A133, A146
\@itemdepth A231, A233, A234, A235
\@itemfudge C40, C46, C73
\@itemitem A235, A238
\@itemlabel A44, A96, A133
\@itempenalty i32, A23, A165
\@iwhiledim f7
\@iwhilenum f3
\@iwhilesw f10
\@ixpt o490
\@ixstackcr D73
\@killglue D30, D38, D44
\@kludgeins K270, K271, K272, K274, K300, K301, K379, K400, K401, K407, K408, K409, K418, K434, K438, K448, K1191, K1230
\@labels A27, A136, A137, A179, A196, A197
\@largefloatcheck G117, G134, G154
\@lastchclass C232, C242, C243, C245, C253, C276, C290, C294, C303, C316, C317
\@latex@e@error d292
\@latex@error I47, d97, d124, g150, g191, g215, g222, g225, g229, g232, g234, g238, g241, g244, g248, g254, g261, g266, g269, g272, g274, g276, g280, g285, g288, k88, l52, o8, o33, o77, o119, o162, o201,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

o256, p105, q90, q101, r23, r64, r84, r125, r146, r162, r242, r274, r314, r318, r360, r365, r420, r488, r494, r528, r532, r536, r570, r580, r664, r669, r672, r704, r707, r761, r764, r767, r834, r840, s36, s86, v126, y54, y129, y141, z253, A209, C91, C100, F4, G6, L214, L233, L246, L315, L390, L407, L415, L420
`\@latex@info` d197, g150
`\@latex@info@no@line` g150, K458
`\@latex@warning` I22, I45, g150, g188, l60, x14, D259, G158, K1292, L470, L476
`\@latex@warning@no@line` d175, g150, g189, k13, k202, x8, x26, x27, y31, F6, K191, K223, K1165, K1358, L86, L261, L342, L435, L442, L500
`\@latexbug` g271, K285, K1151
`\@latexerr` g188, K182, K314, K1221, K1238
`\@lbibitem` I3, I4
`\@ldots` t412, t414
`\@leftcolumn` K102, K1472, K1481
`\@leftmark` J16, J36
`\@let@token` d257, d260, d263, d271, i172, i173, i180, v66, v79, z153, z155, z158
`\@lign` z138, z140
`\@linechar` . . D90, D91, D92, D96, D97, D99, D104, D106, D107, D108, D109, D111, D115, D116, D119, D120, D125, D150, D325
`\@linefnt` D47, D49, D90, D143, D151, D182, D185, D332
`\@linelen` D78, D79, D103, D110, D119, D121, D126, D127, D128, D136, D137, D178, D181, D183, D184, D326
`\@listctr` I9, A192, A215
`\@listdepth` A23, A35, A38, A43, A99, B218
`\@listfiles` k52, k208, k223
`\@loadwithoptions` . L220, L226, L230
`\@lowpenalty` i55, N3
`\@ltab` C62, C97
`\@m` I17, b21, b150, b152, b153, b186, b187, i190, k39, A80, D113, D117
`\@mainaux` k5, k31, k32, k81, k93, k118, y15
`\@makebox` B7, B9
`\@makecaption` G24
`\@makecol` K209, K336, K356
`\@makefcolumn` K317, K318, K326, K328, K1456
`\@makefnmark` G255, G298
`\@makefntext` B246, G281
`\@makeother` a23, a44, a73, d277, d278, o357, o358, o359, o360, o361, o362, o363, o364, o365, o366, o367, y113, y123, y134, L110, L111, L458
`\@makepicbox` B6, B30, D232
`\@makespecialcolbox` K380, K404
`\@marbox` . G199, G201, G205, G209, G210, G234, K1150, K1160, K1163, K1171, K1173, K1174, K1176, K1177, K1178, K1187
`\@marginparreset` G218, G225
`\@markright` J29, J34
`\@maxdepth` . k50, K72, K362, K394, N9
`\@maxtab` C2, C85
`\@medpenalty` i56, N3
`\@midlist` K47, K375, K376, K769, K771, K880
`\@minipagefalse` A171, B191, B193, B231, G112, G144, G220
`\@minipagerestore` B219, B221
`\@minipagetrue` B192, G111
`\@minus` d13, K1528, K1529, K1530, K1533, K1534
`\@missingfileerror` k167, k174, *365*, L335
`\@miv` e3
`\@mkboth` J11, J13
`\@mklab` A45, A130
`\@mkpream` C171, C204, C232
`\@mparbottom` G242, G243, K99, K352, K1161, K1169, K1170, K1171, K1172
`\@mpargs` B210, B234
`\@mparswitchfalse` K83
`\@mpfn` . B216, G262, G267, G307, G311
`\@mpfootins` B225, B226, B229, B235, B238, B239
`\@mpfootnotetext` B217, B237
`\@mplistdepth` B218, B235
`\@multicnt` C338, C340, C341, C342, C349, C350, C351, D38, D39, D41, D322, D359, D361, D362, D363, D364, D368, D372, D383, D387

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@multiplelabels k27, x25, x31, y29, y35
\@multiput D36, D37
\@multispan C339, C343, C347
\@namedef *23*, d27, k128, l894, o110, o111, o135, p378, x28, y121, z257, z258, C157, E12, E13, E18, E19, E23, E24, E25
\@nameuse *23*, d28, k116, k127, E23, J5, K486
\@nbitem A158, A211
\@ne b16
\@needsf@rmat L256, L259, L264
\@needsformat L244, L254, L258
\@negargfalse D86
\@negargtrue D85
\@newcommand d46, d47
\@newctr m13, m15, E8
\@newenv d119, d120, d130
\@newenva d117, d118
\@newenvb d119, d120
\@newl@bel I10, x22, y17
\@newline i45, i47
\@newlistfalse A29, A33, A108, A172, K477
\@newlisttrue A29, A33, A87
\@next G56, G198, G199, K9, K161, K261, K668, K1150
\@nextchar C239, C240, C298, C299, C300
\@nil a108, a109, c5, c11, d30, d31, d32, d104, d251, d252, f13, f19, f27, j14, l82, o275, o288, o373, o433, o436, o437, o445, p306, p307, p309, p322, p328, p332, p333, p373, p394, p401, p509, p523, q16, q34, q43, q47, r40, r284, r292, r325, r845, r847, v41, v45, C335, C336, L27, L29, L60, L61, L67, L195, L198, L292, L300
\@nmbrlistfalse A33, A46, A91
\@nmbrlisttrue A215
\@nnil f13, f20, f21, f22, f28, o185, p133, p135, p299, p301, p315, p317, p322, p336, p338, p345, p360, p361, p363, p394, p401
\@no@font@optfalse q119
\@no@lnbk i13, i14, i15
\@no@pgbk i3, i4, i5
\@nobreakfalse i58, i60, A183, F77, F112, F140, G107, K142, K889
\@nobreaktrue i59, F109, G106
\@nocnterr g224
\@nocounterr g224, m4, m8, m16, E21
\@nodocument g231, k58, y50, G35, K133, K160
\@noitemargfalse A32, A190
\@noitemargtrue A32, A133
\@noitemerr g275, i111, i130, A69, A81, A107
\@noligs y114, y135, y151
\@nolnerr g221, i17, i51, y68
\@nomath o2, o254, s31, s49, s51, s56
\@noparitemfalse A30, A135
\@noparitemtrue A30, A66
\@noparlistfalse A31, A70
\@noparlisttrue A31, A67
\@normalcr i35, i43, B190
\@normalsize L4, L5
\@noskipsecfalse k45, F81, K135
\@noskipsectrue F21, F78
\@notdefinable d105, d106, d110, g214
\@notprerr g279, k56
\@nthm E3, E4
\@nxttabmar C11, C23, C25, C27, C66, C102, C103, C109, C110
\@obsoletefile k201
\@oddfoot J11, J14, J15, K105, K489
\@oddhead J11, J14, K104, K489
\@onefilewithoptions L284, L288, L294, L304, L353
\@onelevel@sanitize d279, G38
\@onlypreamble I40, d33, d161, d163, d172, d180, k61, k70, k85, k203, k229, l23, l24, l66, l67, l71, l94, l114, l144, l145, l159, l895, o26, o90, o92, o98, o114, o142, o157, o178, o183, o198, o384, p379, q18, q26, q32, q69, q73, q78, q83, q88, q98, q116, q117, q118, q124, q128, q133, r17, r19, r44, r46, r72, r81, r106, r176, r177, r180, r212, r245, r247, r249, r262, r277, r324, r326, r368, r407, r423, r500, r539, r542, r583, r586, r589, r609, r622, r675, r710, r714, r717, r770, r790, r794, r858, 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, L82, L91, L99, L101, L118, L121, L122, L133, L134,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

L135, L142, L148, L161, L174, L186, L188, L193, L199, L204, L208, L211, L219, L224, L227, L231, L240, L253, L258, L264, L273, L278, L303, L353, L355, L364, L377, L378, L381, L388, L397, L404, L405, L413, L418, L423, L506, L507, L508, L509, L511
\@opargbegintheorem E32, E35
\@opcol K210, K218, K318, K336, K341
\@options L187
\@othm E3, E20
\@outerparskip A8, A88, A117, A142, A212
\@outputbox K101, K359, K361, K383, K386, K387, K411, K413, K414, K419, K422, K427, K429, K436, K442, K444, K515, K541, K547, K557, K558, K581, K588, K651, K654, K657, K663, K664, K1472, K1478, K1486
\@outputdblcol K344, K1469
\@outputpage K327, K346, K472, K1493, K1501
\@oval D263, D264
\@ovbtrue D265
\@ovdx D239, D273, D279, D281, D292, D294, D348, D349, D350, D351, D365, D366, D368, D382
\@ovdy D239, D274, D280, D281, D286, D290, D355, D356, D357, D358, D369, D370, D372, D386
\@ovhorz D278, D279, D291
\@ovltrue D265
\@ovri B17, D239, D272, D286, D295
\@ovro D239, D272, D279, D280, D285, D290, D291, D300, D307
\@ovrtrue D265
\@ovttrue D265
\@ovvert D276, D277, D283
\@ovxx D239, D267, D269, D273, D277, D278, D291, D345, D346, D347, D351, D360, D361, D367, D368, D381
\@ovyy D239, D268, D269, D274, D279, D283, D352, D353, D354, D358, D360, D371, D372, D385
\@p@pfilename L27, L29, L34
\@pagedp K98, K258, K263, K831, K1179, K1189
\@pageht K97, K259, K263, K265, K266, K267, K272, K830, K1162, K1169
\@par h3, h6
\@parboxrestore B162, B190, B215, B242, G19, G96, G217, G276, K167, K478
\@parboxto B157
\@parmoderr g265, G54, G195
\@parse@version L60, L61, L67, L68
\@partaux k5, k87, k103, k105, k106, k112, k121, k123, k126
\@partlist k84, k99
\@partswfalse k8
\@partswtrue k83
\@pass@ptions L113, L118, L119, L120, L329
\@pboxswfalse B160, B208
\@pboxswtrue B170
\@penup z129, z130
\@percentchar a53, L450, L452, L454, L456, L495
\@picbox D6, D21, D27, D28
\@picht D6, D20, D27
\@picture D12, D18
\@picture@warn D123, D251, D255, D259
\@pkgextension L16, L40, L51, L69, L119, L210, L213, L230, L295, L399
\@plus d13, i197, F16, F151, J40, K1528, K1529, K1530, K1533, K1534, K1538, K1539, K1540, K1544, K1545, K1546
\@pnumwidth F163
\@popfilename L20, L350
\@pr@videpackage L90, L92, L99
\@preamble C172, C174, C182, C207, C226, C228, C229, C233, C248, C266, C267, C302
\@preamblecmds d33, k57, L518, L519
\@preamerr g250, C181, C244, C323
\@process@pti@ns L160, L173, L175, L186
\@process@ptions L147, L149, L161
\@protected@testopt d57, d69
\@providesfile a45, a46, L102, N235
\@ptionlist L37, L74, L146, L319, L325, L400
\@pushfilename L20, L305
\@put D262, D281, D307
\@qend d105, d251, g218
\@qrelax d106, d251
\@rc@ifdefinable d99, d101, d212, l14
\@reargdef d91
\@refundefined k46, x3, y27

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@reinserts K279, K282, K397
\@removeelement f32, L190
\@reqcolroom K830, K831, K834, K836, K837, K842, K846, K848, K875, K876, K977, K979, K981, K984, K986, K1282, K1399, K1403, K1406
\@reset@ptions L310, L351, L356
\@resetactivechars K457, K475
\@resethfps K944, K1349
\@restorepar h6, i139, i155, *46*, A126
\@reversemarginfalse G243, K82
\@reversemargintrue G242
\@rightmark J16, J37
\@rightskip y79, y83, A75, B186
\@rjfieldfalse C36, C68
\@rjfieldtrue C116
\@roman m35, m41
\@rsbox B266, B267
\@rtab C62, C77
\@rule B250, B251
\@sanitize d277, H7, H18, H24, H35
\@savebox B57, B60
\@savemarbox G205, G206, G209, G212
\@savepicbox B57, B64
\@savsf i61, i67, i72, i80
\@savsk i61, i66, i73, i81
\@scolelt K606, K648
\@sdblcolelt K622, K649
\@seccntformat F43, F94
\@secondoftwo a35, d184, d249, k149, l100, l913, l929, x21, J17, L46, L62, L80
\@secpenalty i33, F19, F33
\@sect F37, F38
\@seqncr z256
\@setckpt k121, k128, y16
\@setfloattypecounts K814, K962, K1038, K1296
\@setfontsize s56
\@setfps G34
\@setfpsbit G70, G73, G76, K1340
\@setminipage B220, G21, G102, G110, G231
\@setnobreak G104, G230
\@setpar h3, *46*, A78
\@setref x10
\@setsize s56
\@settab C62, C84
\@settodim n6
\@settopoint n11
\@sharp C178, C205, C235, C250, C251, C269, C271, C273, C301
\@shipoutsetup K472
\@shortstack D63, D64
\@sline D81, D84, D147
\@slowromancap m42, m43
\@spaces g192
\@specialoutput K204
\@specialpagefalse K78, K486
\@specialpagetrue J9
\@specialstyle J9, K486
\@sptoken d260, d270
\@sqrt z203
\@ssect F36, F95
\@stackcr D70, D73
\@star@or@long d39, d44, d93, d115, d121, d151, d160, d194
\@startcolumn K211, K218, K593
\@startdblcolumn K593, K1499, K1505
\@startfield C30, C48, C83, C95, C116, C124
\@startline C22, C59, C60, C61, C74
\@startpbox C175, C206, C236, C300, C352, C354
\@startsection F22
\@starttoc F132
\@stopfield C34, C50, C61, C77, C84, C116, C118, C127, C129
\@stopline C32, C58, C76
\@stpelt m20, m23
\@strip@args l79
\@svector D139, D147
\@sverb y136, y137, y144
\@svsec F40, F43, F49, F61
\@svsechd F59, F84, F104
\@sxverbatim y95, y121
\@tabacckludge l178, l180, l353, l354
\@tabacol C160, C228
\@tabarray C152, C162, C163
\@tabclassiv C162, C298
\@tabclassz C161, C252
\@tabcr C58, C64
\@tabfbox C16, C71, C73
\@tablab C63, C117
\@tabminus C63, C108
\@tabplus C63, C101
\@tabpush C11, C68, C76, C127, C130, C132
\@tabrj C63, C115
\@tabular C156, C159, C160
\@tabularcr C162, C190

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@tempboxa e13, l74, n6, n7, A195, A201, A202, A204, B13, B14, B15, B16, B21, B22, B23, B24, B100, B116, B123, B133, B211, B234, B273, B274, B275, B282, B283, B284, B285, D182, D183, D257, D258, D272, D275, D280, D281, D300, D301, D306, D307, D373, D391, F121, F122, G201, G235, K255, K301, K306, K307, K448, K505, K512, K513, K539, K543, K555, K561, K568, K569, K570, K571, K575, K583

\@tempcnta e7, r591, r592, r593, r594, r598, C212, C213, C214, C215, D87, D88, D114, D115, D116, D129, D130, D131, D132, D134, D135, D148, D149, D154, D156, D157, D158, D159, D160, D163, D165, D166, D167, D168, D169, D170, D171, D172, D173, D174, D204, D205, D206, D207, D208, D226, D227, D228, D229, D230, D231, D248, D249, D250, D252, D254, D256, D258, D284, D288, D302, D303, D304, D305, D311, D312, D313, D314, D315, D316, D364, D380, G58, G64, G66, G79, G80, G86, G87, K16, K18, K20, K699, K700, K701, K702, K724, K727, K760, K763, K871, K1041, K1044, K1151, K1153, K1156, K1158, K1160, K1182, K1330, K1331, K1335, K1341, K1345, N17, N22, N23, N24, N86, N91, N92, N93

\@tempcntb e7, r592, r596, r598, D157, D158, D159, D161, D162, D163, D284, D285, D288, D289, G84, G85, G86, K17, K20, K21, K1341, K1342, K1343, N18, N22, N87, N91

\@tempdima e10, z116, z119, z125, B27, B28, B115, B116, B121, B122, B123, B125, B161, B162, B209, B213, B254, B257, B258, B271, B273, B279, B282, C37, C38, C39, C79, C80, C81, C82, C200, C201, D110, D111, D113, D114, D115, D116, D117, D118, D247, D248, D249, D258, D273, D274, D276, D277, D303, D305, D310, D311, D312, F156, F157, F166, G121, G123, G156, G157, G158, K177, K178, K179, K363, K365, K417, K419, K420, K425, K430, K434, K439, K443, K686, K689, K704, K714, K1103, K1104, K1107, K1108, K1161, K1162, K1163, K1164, K1167, K1170, K1173, K1175, K1447, K1448, K1450, K1451

\@tempdimb e10, o455, o459, p133, p134, p399, p432, p433, p442, p443, p447, p469, p472, p475, p477, B164, B165, B255, B258, B272, B274, B280, B283, D111, D112, D269, D270, D271, D298, D299, D308, D309, K704, K705, K706, K707, K714

\@tempdimc e10, p426, p427, p429, p430, p432, p433, B256, B257, B258

\@tempskipa e14, i19, i22, i23, p135, p136, A116, A117, A118, A140, A142, A143, A144, A212, A213, A214, F25, F27, F28, F33, F45, F46, F71, F72, F74, F86, F87, F96, F97, K1219, K1220, K1222, K1230

\@tempskipb e14, i87, i89, i91, i94, i96, i106, i122, i125

\@tempswafalse I13, a25, k97, o69, r214, r264, r328, r409, r833, r839, y18, y105, K730, K766, K1047, L431

\@tempswatrue I13, a26, k95, k100, o72, r217, r267, r331, r412, r796, y42, y110, K1049, K1072, K1408, K1425, L430

\@temptokena e16, y45, y46, J22, J23, J30, J31, J34, J35

\@testdef y17, y40

\@testfalse K12, K14, K15

\@testfp K673, K703, K1333, K1459

\@testopt d20, d46, d67, d71, d117, i3, i4, i13, i14, z243

\@testpach C240, C316

\@testpatch C316

\@testtrue K13, K21, K675, K708, K1337

\@text@composite l79

\@text@composite@x l79

\@textbottom J40, J42, K389, K431, K445, K454

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\@textfloatsheight K352, K827, K829, K878, K879, K884, K1282
\@textmin G180, G181, K93, K829, K833, K836, K837, K981, K1065, K1067, K1083, K1390, K1392, K1394
\@textsuperscript G256, G258, G259
\@texttop J40, J42, K385, K412, K454
\@tf@r f25, f26
\@tfor f25, k150, k210, v71, B36, C238, D266, G59
\@tforloop f27, f28, f30
\@thanks F10, F13
\@thefnmark B244, G255, G256, G263, G268, G278, G287, G292, G303, G308
\@thefoot K105, K489, K492, K519
\@thehead K104, K489, K491, K509
\@themargin K55, K490, K492, K504
\@themark J21, J22, J29, J30, J35, J38
\@thirdofthree d188, l152
\@thm E12, E18, E24, E26
\@thmcounter E11, E17, E33
\@thmcountersep E10, E33
\@title F3
\@tocrmarg F152
\@toodeep g240, A36, A222, A233
\@toplist K45, K308, K309, K534, K540, K550, K551, K767, K779
\@topnewpage K147
\@topnum G169, K86, K764, K765, K779, K783, K791, K1024, K1029, K1255, K1276
\@toproom G171, K87, K767, K779, K1256
\@topsep A1, A71, A73, A161
\@topsepadd A1, A59, A61, A71, A124
\@totalleftmargin y102, A9, A53, A54, B185, C37, C67, C72
\@tracemessage K1244
\@traceval K1244
\@trivlist A48, A57, A92
\@tryfcolumn K596, K613, K627, K1460
\@trylist K636, K639, K668, K688
\@twoclasseserror L201, L419
\@twocolumnfalse K80, K128
\@twocolumntrue K154
\@twoloadclasserror L349, L414
\@twosidefalse K81
\@typein d19, d20, d26
\@typeset@protect d70, d217, d224, d226, l26, l32, l165, l173, s57
\@uclclist l815, l816, l863, N165
\@undefined I33, a15, a16, a55, a56, a57, a78, a86, a94, a101, a152, a156, a182, a189, a249, a250, b86, b89, d196, g31, k51, k52, k137, l150, l152, o310, o401, o469, v105, D338, G5, L4, L339, L365, L482, L485, L499, N189, N224, N225, N226, N227, N228
\@unexpandable@noexpand d192
\@unexpandable@protect d192, d229, d235, d240, k75, C234
\@unknownoptionerror L360, L389, L402
\@unprocessedoptions L185, L229, L336, L340, L404
\@unused d4, g15, g35, g62, k3, L504
\@unusedoptionlist k12, k14, L11, L138, L139, L191
\@upline D175, D176, D182
\@upordown D95, D96, D104, D125, D151
\@upvector D146, D182
\@use@ption L156, L168, L178, L180, L189
\@use@text@encoding l115, l1106
\@vbsphack i86
\@verb y136, y144
\@verbatim y100, y118, y121
\@vereq t365, t366
\@viiipt o489
\@viipt o488
\@vipt o487
\@vline D80, D175
\@vobeyspaces y93, y118, y144
\@vpt o486
\@vspace i132
\@vspacer i132
\@vtryfc K642, K650
\@vvector D138, D146
\@warning g188
\@wckptelt k122, k125
\@whiledim f7, D44, D103
\@whilenoop f3
\@whilenum f3, C214, D39, D205, D207, D227, D230, D380
\@whilesw f10, K212, K318, K327, K1500
\@whileswnoop f10
\@wholewidth B85, B87, B88, B90, B92, B93, B94, B95, D2, D48, D50, D57, D177, D180, D218, D225, D287, D293, D334, D335, D377
\@width b193, d13, i193, l245, l248, p146, t522, B90, B92,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

B129, B136, B258, B288, C170, C201, C315, C334, D127, D177, D180, D197, D204, D218, D225, D287, D377, G250, K1189, K1483
\@wrglossary H25, H30
\@wrindex H8, H13
\@writeckpt k110, k119
\@writefile k26, y43, F147
\@writesetup K472
\@wrong@font@char . . l126, o402, o413
\@wtryfc K652, K662
\@x@protect d73, d216
\@x@sf G297, G299
\@xDeclareMathDelimiter . . r621, r676
\@xaddvskip i86, i107
\@xarg D77, D80, D85, D89, D90, D126, D128, D133, D134, D138, D144, D152, D319
\@xargarraycr C187, C196, C200
\@xargdef d47
\@xarraycr C184, C185
\@xbitor K15, K17
\@xcentercr y69, y70
\@xdblarg d275
\@xdblfloat G162
\@xdim D34, D40, D42, D323, D381, D382, D383, D384, D390
\@xeqncr z235
\@xexnoop C208, C218
\@xexpast C209, C210
\@xfloat G28, G29, G34, G164
\@xfootnote G262, G265
\@xfootnotemark G285, G289
\@xfootnotenext G302, G305
\@xhline C328, C329
\@xifnch d261, d271
\@xiipt o493, t83, t85, t86
\@xipt o492, t82
\@xivpt o494, t84, t86
\@xmpar G203, G204
\@xnewline i39, i40, i44
\@xnext K10, K11
\@xnthm E5, E6
\@xobeysp i182, y94, y95
\@xprocess@ptions . L147, L162, L174
\@xpt o491, t81, t84, t85
\@xsect F69, F70, F106
\@xtabcr C58, C59
\@xtabularcr C191, C192
\@xthm E28, E29
\@xtryfc K639, K667
\@xtypein d20, d21
\@xverbatim y95, y118
\@xviipt o495, t85, t87
\@xxDeclareMathDelimiter . . r606, r610
\@xxpt o496, t86, t87
\@xxvpt o497, t87
\@xxxii e2, l324, l326, G85, K670, K671, K700, K701, K1299
\@xympar G207, G211, G233
\@yarg D77, D81, D85, D86, D95, D133, D139, D146, D148, D175, D319
\@yargarraycr C188, C198, C202
\@yargd@f d75
\@yargdef d51, d62, d75, d92
\@ydim D35, D40, D42, D324, D385, D386, D387, D388, D389
\@yeqncr z235
\@ympar G203, G208
\@ynthm E5, E14
\@ythm E28, E29
\@ytryfc K680, K682
\@yyarg D85, D86, D87, D90, D152, D319
\@ztryfc K687, K698
\[. o363, z170, z195, z276, N117
\\ a21, a194, a195, a196, a197, a200, a207, a208, a209, a210, a213, a220, a221, a222, a223, a226, a233, a235, a236, a239, a242, b13, d191, d277, g274, i35, i204, k209, k224, l420, o351, t170, *48*, y76, y83, y89, y97, z217, z311, B190, B272, B274, C64, C152, C162, C176, D70, N102
\{ a3, a7, a21, b2, b13, g25, l262, l422, o352, t168, y96, z59, z108, N105
\} a8, a21, b3, b13, g24, l263, l423, o353, t169, y96, z59, N106
\] . . b239, o364, z170, z196, z286, N118
\^ a10, a19, a22, a66, a253, b7, b9, b11, b14, b158, b159, b173, b174, d5, d278, i204, i206, i208, l186, l241, l284, l355, l362, l418, l501, l508, l512, l517, l522, l527, l534, l540, l541, l547, l552, l597, o349, o350, o355, L427, L428, L429, L481, L484, L487, N49, N50, N51, N52, N54, N55, N56, N57, N59, N103, N109, N110, N111, N112, N125, N126, N127, N154, N155, N156, N157, N159, N160, N161, N162, N164

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

_ a22, b8, b14, d278, l268, t173, z166, z167, N104
\` l187, l285, l319, l353, l360, l437, l499, l506, l510, l515, l520, l525, l532, l536, l537, l545, l550, l598, l637, o365, s154, y145, B181, C63, N119
\| l421, m59, t479, N120
\~ a22, b10, b14, d278, g23, i184, l194, l242, l286, l363, l419, l502, l514, l518, l528, l544, l548, l599, y139, y149, N107

\␣ I17, a21, a38, b13, b158, b176, d277, d291, g22, g23, g24, g25, g28, i183, o348, o480, q11, t171, y93, y94, E36, E38, L104, N97

A

\A N44, N122, N149
\a l178, C1, N33, N123, N138
\AA b164, l195, l327, l389
\aa b164, l200, l321, l399
\abovedisplayshortskip b139, z319
\abovedisplayskip b138, z312, z314, z316, z317, z318, z319
\accent l76, l295, l322, l375, l610
\accent@spacefactor l75, l76, l77
\active a11, a66, a253, b10, b11, b173, b174, b176, y93, y94, y138, y147, z151, z166, K466, L427, L428, L429, L481, L484, L487
\active@math@prime z150, z151, K470
\acute t424
\add@accent l70, l72
\addcontentsline F53, F63, F142, G16
\addpenalty i113, A124, A160, A165, F33, K290, K892
\addto@hook o127, o129, o485, r196, r287, r291, r308, r432, r438, r446, r462, r465, r468, r805, r812, r815
\addtocontents F143, F144
\addtocounter m6, m18, *103*
\addtolength n5, z314, z316, *106*
\addtoversion q130
\addvspace i100, y70, A124, A161, A162, A166, A214, F33
\adjdemerits b118
\AE l196, l300, l390, l615, N178
\ae l201, l303, l400, l619, N178
\afterassignment b189, b192, d230, d236, l167, l175, o245, z129
\aftergroup o66, o259, p156, p222, r79, r91, r99, v47, y142, B73, K473, K481, K482
\aleph t227
\alloc@ b47, b48, b49, b50, b51, b53, b54, b55, b56, b57, o23
\allocationnumber b37, b59, b60, b61, b67, b68, b69, C4, C9
\allowbreak b196, z40
\Alph m38, *103*
\alph m37, *103*
\alpha t187
\alpha@elt r45, r200, r382, r484, r804, r805
\alpha@list r41, r43, r209, r370, r382, r427, r482, r483, r800, r806, r807
\amalg t293
\and *282*, F14
\angle t243
\approx t333
\arabic m34, *103*, E33
\arccos z13
\arcsin z10
\arctan z16
\arg z26
\array C150
\arraycolsep z220, z221, z324, z325, C228, C306
\arrayrulewidth C292, C306, C314, C315, C327, C331, C334, C344, C346
\arraystretch C168, C169, C310
\Arrowvert t475
\arrowvert t473
\ast t151, t309
\asymp t357
\AtBeginDocument I34, I48, k47, L374
\AtBeginDvi K67
\AtEndDocument y9, L374
\AtEndOfClass z275, L374
\AtEndOfPackage L185, L229, L374
\atopwithdelims z57, z58, z59
\aut@global d54, d86, d127, d137, d141, d147, d202, d285, d289
\author *282*, F5

B

\b l188, l291, l371, l606
\backslash t170, t494
\badness b97
\bar t428

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\baselineskip b157, b187, b223, p140, p141, p142, p144, p145, t418, z112, z113, z121, z127, z131, B188, C180, D67, D187, K190, K221, K501, K516
\baselinestretch o236, p118, p119, p138, p199
\batchmode k186, k187, q96, s121, N198, N219
\begin g232, g234, l563, p7, t4, u4, y51, y52, z279, F14, F17, G150, G152, L238, M3
\belowdisplayshortskip . . b141, z318
\belowdisplayskip b140, z317
\best@size . . . p400, p434, p440, p446
\beta t188
\bezier *260*, D341, D342
\bfdefault s14, t32
\bfseries I20, s12, s13, v19, x13, E36, E38
\bgroup b171
\bibcite I7, I9, I10
\bibdata I25, I29
\bibitem I3
\bibliography I27, *307*
\bibliographystyle I32, *307*
\bibstyle I25, I37
\Big t525, z44, z45, z46
\big t524, z41
\bigbreak b203
\bigcap t263
\bigcirc t306
\bigcup t264
\Bigg t527, z50, z51, z52
\bigg t526, z47, z48, z49
\Biggl z50
\biggl z47
\Biggm z51
\biggm z48
\Biggr z52
\biggr z49
\Bigl z44
\bigl z41
\Bigm z45
\bigm z42
\bigodot t271
\bigoplus t270
\bigotimes t269
\Bigr z46
\bigr z43
\bigskip b208, i162
\bigskipamount . b207, i164, i165, G246
\bigsqcup t274
\bigtriangledown t279, t280
\bigtriangleup t278, t281
\biguplus t262
\bigvee t260
\bigwedge t261
\binoppenalty b109
\bm@b B21
\bm@c B21
\bm@l B21
\bm@r B21
\bm@s B21
\bm@t B21
\bmod z35
\boldmath j14, s49
\bordermatrix z115
\bot t242
\botfigrule K560, K1547
\botmark J36, K526
\bottomfraction G173, K1516
\bowtie t390
\Box s92
\boxmaxdepth b132, D264, D298, K362, K384, K428, K533, K542, K582
\brace z59
\braceld . . t459, t463, t464, t466, t468
\bracelu t461, t465, t467
\bracerd t460, t465, t467
\braceru t462, t464, t468
\bracevert t512
\brack z58
\break b196, b201, i53
\breve t429
\brokenpenalty b114
\buildrel t377, z107
\bullet t295
\bx@A K23, K41
\bx@B K24, K41
\bx@C K25, K41
\bx@D K26, K41
\bx@E K27, K41
\bx@F K28, K42
\bx@G K29, K42
\bx@H K30, K42
\bx@I K31, K42
\bx@J K32, K42
\bx@K K33, K43
\bx@L K34, K43
\bx@M K35, K43
\bx@N K36, K43
\bx@O K37, K44

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\bx@P K38, K44
\bx@Q K39, K44
\bx@R K40, K44

C

\c l189, l294, l374, l463, l465, l490, l492, l505, l531, l609
\c@bottomnumber . . G167, G172, K1514
\c@dbltopnumber . . G166, G178, K1521
\c@enumi A217
\c@enumii A217, A217
\c@enumiv A217
\c@equation z197, z230, z331
\c@errorcontextlines g181
\c@footnote F11, G252, G291
\c@mpfootnote B216, G254
\c@ncel t369, t370
\c@page w3, w6, w7, K119, K1156
\c@secnumdepth . . F39, F54, F64, F123
\c@tocdepth F123, F150
\c@topnumber G165, G169, K1510
\c@totalnumber . . . G168, G174, K1517
\cal s155
\calculate@math@sizes . . . o450, p173
\cap t286
\capitalacute l683, l967
\capitalbreve l690, l974
\capitalcaron l689, l973
\capitalcedilla l676, l964
\capitalcircumflex l684, l968
\capitaldieresis l686, l970
\capitaldotaccent l692, l976
\capitalgrave l682, l966
\capitalhungarumlaut l687, l971
\capitalmacron l691, l975
\capitalnewtie l696, l1037
\capitalogonek l679, l965
\capitalring l688, l972
\capitaltie l694, l1035
\capitaltilde l685, l969
\caption G4
\cases z108
\cdot t308
\cdotp t410, t416
\cdots t416
\cdp@elt o71, o91, o102, o103, o124, o127, o129, r134, r216, r266, r330, r411
\cdp@list . . . o73, o89, o103, o131, o132, r152, r218, r268, r332, r413
\center y73
center (environment) y73
\centering y73, y75
\centerline B289
\cf@encoding l34, l41, l44, l53, l119, o204, o214, o224, o243
\ch@ck b58, b63, b64, b65, b66, b70, L438
\changes l560, G151, N71
\char l293, l296, l329, l332, l343, l350, l373, l377, l382, l384, l386, l388, l580, l608, l611, l641, l648, l655, l678, l681, l729, s55, y150, z148, D132, D160, D174, D182, D185, D258, D285, D289, D302, D303, D305, D316
\chardef a11, a17, a18, b10, b16, b17, b18, b19, b20, b51, b54, b55, b56, b68, b217, b218, b219, e2, l18, o23, C4, C9, L437
\chardef@text@cmd l3
\check t430
\check@command d160, d162
\check@icl v9, v27, v32, v38, v46, v53, v55
\check@icr v9, v27, v33, v39, v47, v56, v61
\check@mathfonts j5, l256, o265, o267, p204
\check@nocorr@ v29
\check@range o316, p335, p336
\check@single o317, p334, p359
\CheckCommand d160
\CheckEncodingSubset l906, l962, l963, l1031, l1091, l1093, l1106
\chi t207
\choose z57
\circ t305
\circle D260, D296
\citation I11, I19, I43
\cite I12, *307*
\cl@@ckpt k122, m24
\cl@page w4
\ClassError g88
\ClassInfo g88
\ClassWarning g88
\ClassWarningNoLine g88
\cleaders b237, t454, t457
\cleardoublepage K119
\clearpage k91, k109, y12, y49, K106, K119, K124, K148, K331, K337, K1464
\cline C335

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\clubpenalty b111, k10, k19, A184, A186, F83, F89, F113, F118
\clubsuit t255
\col@number . . . K76, K129, K156, K168
\colon t411
\color@begingroup o476, z87, z103, B13, B47, B101, B212, B245, C49, C53, G280, K367
\color@endbox B47, G147, G223, K172, K510, K520
\color@endgroup . . o481, z87, z103, B13, B47, B59, B80, B103, B232, B248, C51, G283, K371
\color@hbox B47, K507, K517
\color@setgroup B47, B59, B78
\color@vbox B47, G92, G214, G236, K163
\columnsep k21, K62, K150
\columnseprule K63, K1483
\columnwidth k18, k21, k22, k24, B214, B241, G95, G276, K61, K125, K126, K127, K149, K150, K151, K152, K153, K1181, K1183, K1480, K1485
\cong t365
\contentsline F143, F148
\coprod t259
\copyright l240, l270, s75
\cos z12
\cosh z14
\cot z18
\coth z19
\count@ a13, a126, a127, a128, a133, b41, b193, b194, c6, c7, c8, c9, c10, c12, d140, d145, p22, p256, p258, p280, p281, r193, r195, r199, r509, r510, r511, r551, r552, r553, r594, r595, r596, r634, r635, r636, r642, r643, r644, r687, r688, r689, r695, r696, r697, r737, r738, r739, r745, r746, r747, v98, v101, D379, D380, D381, D384, D385, D388, D392, N29, N30, N39, N41, N134, N135, N144, N146
\countdef a13, b37, b38, b39, b41, b47, w3
\cr b167, l955, l958, z118, z122, z225, z255, z328, C174, C185, C192, C201, C202, C345, D72, D74, D75
\crcr . . . b224, l292, l296, l299, l372, l376, l380, l382, l384, l579, l607, l611, l614, l678, l681, l728, l959, s77, 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, C153, C154, D72
\cs l560
\csc z21
\cup t287
\curr@fontshape l135, o63, o280, o290, o294, o296, o387, o393, o396, o402, o409, o411, p92, p100, p121, p449, p473, p519, p536, r156, r161
\curr@math@size o269, p210, p216, p221, p238
\CurrentOption l845, l847, l859, L13, L140, L150, L151, L152, L157, L164, L165, L166, L169, L176, L177, L181, L182, L183, L190, L192, L196, L197, L198, L309, L391, L392, L401, L402
\CYRA l817
\cyra l817, l862
\CYRABHCH l817
\cyrabhch l817
\CYRABHCHDSC l817
\cyrabhchdsc l817
\CYRABHDZE l818
\cyrabhdze l817
\CYRABHHA l818
\cyrabhha l818
\CYRAE l818
\cyrae l818
\CYRB l818
\cyrb l818
\CYRBYUS l819
\cyrbyus l818
\CYRC l819
\cyrc l819
\CYRCH l819
\cyrch l819
\CYRCHLDSC l819
\cyrchldsc l819
\CYRCHRDSC l820
\cyrchrdsc l819
\CYRCHVCRS l820
\cyrchvcrs l820
\CYRD l820
\cyrd l820
\CYRDELTA l820
\cyrdelta l820

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\CYRDJE 1821
\cyrdje 1821
\CYRDZE 1821
\cyrdze 1821
\CYRDZHE 1821
\cyrdzhe 1821
\CYRE 1821
\cyre 1821
\CYREPS 1822
\cyreps 1821
\CYREREV 1822
\cyrerev 1822
\CYRERY 1822
\cyrery 1822
\CYRF 1822
\cyrf 1822
\CYRFITA 1823
\cyrfita 1822
\CYRG 1823
\cyrg 1823
\CYRGDSC 1823
\cyrgdsc 1823
\CYRGDSCHCRS 1823
\cyrgdschcrs 1823
\CYRGHCRS 1824
\cyrghcrs 1824
\CYRGHK 1824
\cyrghk 1824
\CYRGUP 1824
\cyrgup 1824
\CYRH 1824
\cyrh 1824
\CYRHDSC 1825
\cyrhdsc 1825
\CYRHHCRS 1825
\cyrhhcrs 1825
\CYRHHK 1825
\cyrhhk 1825
\CYRHRDSN 1826
\cyrhrdsn 1825
\CYRI 1826
\cyri 1826
\CYRIE 1826
\cyrie 1826
\CYRII 1826
\cyrii 1826
\CYRISHRT 1826
\cyrishrt 1826
\CYRISHRTDSC 1827
\cyrishrtdsc 1827
\CYRIZH 1827
\cyrizh 1827
\CYRJE 1827
\cyrje 1827
\CYRK 1827
\cyrk 1827
\CYRKBEAK 1828
\cyrkbeak 1828
\CYRKDSC 1828
\cyrkdsc 1828
\CYRKHCRS 1828
\cyrkhcrs 1828
\CYRKHK 1829
\cyrkhk 1828
\CYRKVCRS 1829
\cyrkvcrs 1829
\CYRL 1829
\cyrl 1829
\CYRLDSC 1829
\cyrldsc 1829
\CYRLHK 1830
\cyrlhk 1829
\CYRLJE 1830
\cyrlje 1830
\CYRM 1830
\cyrm 1830
\CYRMDSC 1830
\cyrmdsc 1830
\CYRMHK 1830
\cyrmhk 1830
\CYRN 1831
\cyrn 1831
\CYRNDSC 1831
\cyrndsc 1831
\CYRNG 1831
\cyrng 1831
\CYRNHK 1831
\cyrnhk 1831
\CYRNJE 1832
\cyrnje 1831
\CYRNLHK 1832
\cyrnlhk 1832
\CYRO 1832
\cyro 1832
\CYROTLD 1832
\cyrotld 1832
\CYRP 1832
\cyrp 1832
\CYRPHK 1833
\cyrphk 1833
\CYRQ 1833
\cyrq 1833
\CYRR 1833
\cyrr 1833

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\CYRRDSC l833
\cyrrdsc l833
\CYRRHK l834
\cyrrhk l833
\CYRRTICK l834
\cyrrtick l834
\CYRS l834
\cyrs l834
\CYRSACRS l834
\cyrsacrs l834
\CYRSCHWA l835
\cyrschwa l835
\CYRSDSC l835
\cyrsdsc l835
\CYRSEMISFTSN l835
\cyrsemisftsn l835
\CYRSFTSN l836
\cyrsftsn l836
\CYRSH l836
\cyrsh l836
\CYRSHCH l836
\cyrshch l836
\CYRSHHA l836
\cyrshha l836
\CYRT l837
\cyrt l837
\CYRTDSC l837
\cyrtdsc l837
\CYRTETSE l837
\cyrtetse l837
\CYRTSHE l837
\cyrtshe l837
\CYRU l838
\cyru l838
\CYRUSHRT l838
\cyrushrt l838
\CYRV l838
\cyrv l838
\CYRW l838
\cyrw l838
\CYRY l838
\cyry l838
\CYRYA l839
\cyrya l839
\CYRYAT l839
\cyryat l839
\CYRYHCRS l839
\cyryhcrs l839
\CYRYI l839
\cyryi l839
\CYRYO l840
\cyryo l839
\CYRYU l840
\cyryu l840
\CYRZ l840
\cyrz l840
\CYRZDSC l840
\cyrzdsc l840
\CYRZH l840
\cyrzh l840
\CYRZHDSC l841
\cyrzhdsc l841

D

\d l190, l297, l378, l612
\dag l266
\dagger l266, m58, m59, t289
\dashbox D187
\dashv t317
\date *282*, F7
\day a132, L455
\dblfigrule K585, K1547
\dblfloatpagefraction G182, K1524
\dblfloatsep K571, K583, K1106, K1531
\dbltextfloatsep K170, K178, K587, K1105, K1531
\dbltopfraction G179, K1523
\ddag l267
\ddagger l267, m58, m60, t288
\ddot t426
\ddots t421
\deadcycles k115, y39, y49, K249
\declare@robustcommand d194
\DeclareEncodingSubset l890, l897, l898, l899, l900, l1116, l1117, l1118, l1119, l1120, l1121, l1122, l1123, l1124, l1125, l1126, l1127, l1128, l1129, l1130, l1131, l1132, l1133, l1134, l1135, l1136, l1137, l1138, l1139, l1140, l1141, l1142, l1143, l1144, l1145, l1146, l1147, l1148, l1149, l1150, l1151, l1152, l1153, l1154, l1155, l1156, l1157, l1158, l1159, l1160, l1161, l1162, l1163, l1164, l1165, l1166, l1167, l1168, l1169, l1170, l1171, l1172, l1173, l1174, l1175
\DeclareErrorFont o369, r165, s101, t28
\DeclareFixedFont o48
\DeclareFontEncoding l279, l359, l559, l585, l591, l674, o93, t45, t46, t47, t48
\DeclareFontEncoding@ o97, o99, o114
\DeclareFontEncodingDefaults o143, q80, q81, t15

\DeclareFontFamily o68, q75, q76
\DeclareFontShape o27, o29, q15, q17, q71, q72
\DeclareFontShape@ o30, o31
\DeclareFontSubstitution l592, l675, o116, t16, t17, t49, t50, t51, t52
\DeclareMathAccent r501, r539, t424, t425, t426, t427, t428, t429, t430, t431, t432, t433, t434, t435, t436
\DeclareMathAlphabet q109, q113, q115, q122, r327, r490, r501, t70, t71, t72, t73
\DeclareMathAlphabetCharacter . r600
\DeclareMathDelimiter r602, 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 r718, t437
\DeclareMathSizes o180, t76, t77, t78, t79, t80, t81, t82, t83, t84, t85, t86, t87
\DeclareMathSizes* o180
\DeclareMathSymbol r543, r601, r618, t88, t89, t90, t91, t92, t93, t94, t95, t96, t97, t98, t99, t100, t101, t102, t103, t104, t105, t106, t107, t108, t109, t110, t111, t112, t113, t114, t115, t116, t117, t118, t119, t120, t121, t122, t123, t124, t125, t126, t127, t128, t129, t130, t131, t132, t133, t134, t135, t136, t137, t138, t139, t140, t141, t142, t143, t144, t145, t146, t147, t148, t149, t150, t151, t152, t153, t154, t155, t156, t157, t158, t159, t160, t161, t162, t163, t164, t165, t166, t167, t168, t169, t170, t180, t181, t183, t187, t188, t189, t190, t191, t192, t193, t194, t195, t196, t197, t198, t199, t200, t201, t202, t203, t204, t205, t206, t207, t208, t209, t210, t211, t212, t213, t214, t215, t216, t217, t218, t219, t220, t221, t222, t223, t224, t225, t226, t227, t229, t230, t231, t232, t233, t234, t235, t236, t237, t238, t239, t241, t242, t247, t248, t249, t250, t252, t253, t254, t255, t256, t257, t258, t259, t260, t261, t262, t263, t264, t265, t267, t268, t269, t270, t271, t272, t274, t275, t276, t277, t278, t279, t282, t284, t286, t287, t288, t289, t290, t291, t292, t293, t294, t295, t296, t297, t298, t299, t300, t301, t302, t303, t304, t305, t306, t307, t308, t309, t310, t311, t312, t313, t314, t315, t316, t317, t318, t319, t320, t321, t322, t323, t324, t325, t327, t329, t331, t332, t333, t334, t335, t336, t337, t338, t339, t340, t341, t343, t344, t345, t346, t347, t349, t351, t353, t354, t355, t356, t357, t358, t359, t360, t361, t362, t363, t385, t387, t409, t410, t411, t459, t460, t461, t462, t518, t519, t520
\DeclareMathVersion r178, s1, s2
\DeclareOldFontCommand . . v108, v124
\DeclareOption . . . l844, l897, l898, l899, l900, l901, l903, p29, p37, p45, p53, p56, p60, *364*, L123, L412
\DeclareOption* *364*, L123
\DeclarePreloadSizes o160, q85, q86, u19, u21, u22, u23, u25, u26, u27, u28, u29, u30, u34, u38, u43, u45, u49, u50, u53, u54, u57, u58, u64
\DeclareRobustCommand I12, d194, g4, g11, g33, g60, i35, i43, i132, i168, i182, i187, i191, j3, j13, l261, l262, l263, l264, l265, l266, l267, l268, l270, l272, l274, l1101, o199, o227, o228, o229, o233, o235, o253, p113, s3, s6, s9, s12, s15, s18, s21, s24, s27, s30, s75, s79, t364, t368, t371, t376, t378, t380, t383, t389, t391, t393, t395, t397, t399, t401, t403, t405, t407, t413, t415, t417, t420, v3,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

v109, z203, z260, G257, N165, N172
\DeclareSizeFunction . p377, p464, p465, p480, p481, p489, p490, p502, p503, p531, p532, p543, p544
\DeclareSymbolFont q126, r213, t60, t61, t62, t63
\DeclareSymbolFontAlphabet r791, t67, t68, t69
\DeclareSymbolFontAlphabet@ r792, r795
\DeclareTextAccent l69, l280, l281, l282, l283, l284, l285, l286, l287, l288, l289, l290, l360, l361, l362, l363, l364, l365, l366, l367, l368, l369, l370, l588, l593, l594, l595, l596, l597, l598, l599, l600, l601, l602, l603, l682, l683, l684, l685, l686, l687, l688, l689, l690, l691, l692, l693, l694, l695, l696
\DeclareTextAccentDefault l140, l181, l182, l183, l184, l185, l186, l187, l188, l189, l190, l191, l192, l193, l194, l234, l237, l964, l965, l966, l967, l968, l969, l970, l971, l972, l973, l974, l975, l976
\DeclareTextCommand . . l3, l63, l70, l291, l294, l297, l313, l314, l321, l323, l325, l327, l333, l335, l337, l344, l371, l374, l378, l381, l383, l385, l387, l577, l604, l606, l609, l612, l642, l649, l676, l679, l726
\DeclareTextCommandDefault l62, l141, l143, l238, l241, l242, l243, l244, l246, l250, l254, l255, l257, l258, l259, l260, l1030, 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, l1071, l1072, l1073, l1074, l1075, l1076, l1077, l1078, l1079, l1080, l1081, l1082, l1083, l1084, l1085, l1086, l1087, l1088, l1089, l1090, l1092
\DeclareTextComposite l79, l351, l352, 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, l471, l472, l473, l474, l475, l476, l477, l478, l479, l480, l481, l482, l483, l484, l485, l486, l487, l488, l489, l490, l491, l492, l493, l494, l495, l496, l497, l498, l499, l500, l501, l502, l503, l504, l505, l506, l507, l508, l509, l510, l511, l512, l513, l514, l515, l516, l517, l518, l519, l520, l521, l522, l523, l524, l525, l526, l527, l528, l529, l530, l531, l532, l533, l534, l535, l536, l537, l538, l539, l540, l541, l542, l543, l544, l545, l546, l547, l548, l549, l550, l551, l552, l553, l554, l656, l657, l658, l659, l660, l661, l662, l663, l664, l665, l666, l667, l668, l669, l670, l671
\DeclareTextCompositeCommand l79, l330, l353, l354, l355, l356, l555, l556, l639
\DeclareTextFontCommand v1, v15, v16, v17, v18, v19, v20, v21, v22, v23, v24, v25, v123
\DeclareTextSymbol l3, l300, l301, l302, l303, l304, l305, l306, l307, l308, l309, l310, l311, l312, l315, l316, l317, l318, l319, l320, l389, l390, l391, l392, l393, l394, l395, l396, l397, l398, l399, l400, l401, l402, l403, l404, l405, l406, l407, l408, l409, l410, l411, l412, l413, l414, l415, 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, l564, l565, l566, l567, l568, l569, l570, l571, l572, l573, l574, l575, l576, l586, l587, l615, l616, l617, l618, l619, l620, l621, l622, l623, l624, l625, l626, l627, l628, l629, l630, l631, l632, l633, l634, l635, l636, l637, l638, l697, l698, l699, l700, l701, l702, l703, l704, l705, l706, l707, l708, l709, l710, l711, l712, l713, l714, l715, l716, l717, l718, l719, l720, l721, l722, l723, l724, l725, l732, l733, l734, l735, l736, l737, l738, l739, l740, l741, l742, l743, l744, l745, l746, l747, l748, l749, l750, l751, l752, l753, l754, l755, l756, l757,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

l758, l759, l760, l761, l762, l763, l764, l765, l766, l767, l768, l769, l770, l771, l772, l773, l774, l775, l776, l777, l778, l779, l780, l781, l782, l783, l784, l785, l786, l787, l788, l789, l790, l791, l792, l793, l794, l795, l796, l797, l798, l799, l800, l801, l802, l803, l804, l805, l806, l807, l808, l809, l810, l811
\DeclareTextSymbolDefault . . . l140, 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, l229, l230, l231, l232, l233, l235, l236, l977, l978, l979, l980, l981, l982, l983, l984, l985, l986, l987, l988, l989, l990, l991, l992, l993, l994, l995, l996, l997, l998, l999, l1000, l1001, l1002, l1003, l1004, l1005, l1006, l1007, l1008, l1009, l1010, l1011, l1012, l1013, l1014, l1015, l1016, l1017, l1018, l1019, l1020, l1021, l1022, l1023, l1024, l1025, l1026, l1027, l1028, l1029
\default@ds . . . L132, L143, L179, L358, L360
\default@family . . . o104, o136, o375, o378, o397
\default@M . . . o111, o151, o154, o158
\default@mextra . . . q1, q79
\default@series . . . o104, o137, o376, o379, o394
\default@shape . . . o105, o138, o377, o380, o392
\default@T . . . o145, o148, o158, o220
\defaulthyphenchar . . . b121
\defaultscriptratio . . . o455, o466
\defaultscriptscriptratio . . . o456, o466
\defaultskewchar . . . b122
\define@mathalphabet . . . q1, q121
\define@mathgroup . . . q1, q125
\define@newfont . . . o272, o282
\deg . . . z34
\delcode . . . r716
\delimiter . . . r647, r712
\delimiterfactor . . . b123
\delimitershortfall . . . b133
\Delta . . . t217
\delta . . . t190
\depth . . . B16, B19
\det . . . z30
\DH . . . l391, N179
\dh . . . l401, N179
\Diamond . . . s93
\diamond . . . t294
\diamondsuit . . . t256
\dim . . . z28
\dimen@ . . . b41, b190, b191, b227, b228, b230, b232, g31, g32, i147, i152, l328, l329, l331, l332, l640, l641, l956, l958, o185, o188, o192, o454, o455, o456, o460, p423, p424, p425, p426, p430, z72, z73, z129, z130, z131, z132, B281, B284, C158, C159, K386, K388, K413, K415
\dimen@i . . . b41
\dimen@ii . . . b41
\dimendef . . . b42, b43, b44, b48
\discretionary . . . d10, z148
\displ@y . . . z134, z138, z139
\displaylines . . . z133
\displaymath . . . z195
displaymath (environment) . . . z193
\displaystyle . . . t440, t443, t446, t448, z62, z140, z219, z222, z259, z283, z299, z323, z326
\displaywidowpenalty . . . b113
\displaywidth . . . z140, z218, z271, z302
\div . . . t297
\DJ . . . l392, N179
\dj . . . l402, N179
\do . . . I16, I41, a21, a22, a73, b13, b14, d36, f3, f7, f16, f26, k56, k59, k99, k151, k210, k216, o310, o311, o312, o313, o314, o315, o316, o317, o318, o319, o320, o321, o322, o323, o324, o325, o326, v73, y113, y134, y145, y151, B36, C214, C239, D39, D44, D103, D206, D208, D228, D231, D266, D380, G61, L78, L150, L164, L176, L181, L196, L401, L458, L517
\do@noligs . . . y146, y151
\do@subst@correction . o58, p454, p527
\DocInput . . . p8, t5, u5, M4
\document . . . I40, k11, *60*
\document@select@group . . . r107, r169
\documentclass . . p2, t2, u2, L200, L207, L234, L237, L326, L421, M2

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\documentstyle L205, L421
\dorestore@version r79, r89
\dospecials a21, a73, b13, y113, y134, L458
\dot t433
\doteq t377
\dotfill b234
\dots l274, l276
\doublehyphendemerits b116
\doublerulesep C279, C306, C330
\Downarrow t490
\downarrow t484
\downbracefill t445, t463
\ds@ L145, L362
\dt@pfalse z135
\dt@ptrue z134
\dump N243

E

\E L459, L462, L489
\egroup b171
\eject b201
\ell t231
\em s30, s31, v25
\emergencystretch b96, J45, J51
\emph v25
\empty b169
\empty@sfcnt o321, p464, p465, p468, p486, p495, p539
\emptyset t238
\enc@update o205, o207, o223, o226, p129
\encodingdefault l845, l871, r170, s80, t38
\end a16, d8, d251, g235, p9, t6, u6, y60, y97, y98, z290, A112, F15, F17, L467, L471, L477, M5
\end@dblfloat G130
\end@float G114, G139, G238
\endarray C153
\endcenter y74
\enddisplaymath z196
\enddocument y8
\endenumerate A230
\endeqnarray z227, z258
\endequation z199
\endfilecontents L425
\endflushleft y81
\endflushright y87
\endgraf b166
\enditemize A241
\endline b166, z118
\endlinechar . . a39, a40, a41, a151, d23, d25, k182, L105, L106, L107
\endlist A98, A230, A241
\endlrbox B80
\endmath z194
\endminipage B222
\endpicture D25
\endsloppypar J49
\endtabbing C75
\endtabular C153
\endtabular* C153
\endtrivlist y74, y81, y87, y119, z304, A100, A101, C76, E39
\endverbatim y118, y122
\enlargethispage K1196
\enlargethispage* K1196
\enskip i201
\enspace i198
\ensuremath m58, z260, G260
\enumerate A221
enumerate (environment) A221
environments:
  center y73
  displaymath z193
  enumerate A221
  eqnarray z205, z305
  eqnarray* z256
  equation z197, z293
  filecontents *362*
  flushleft y80
  flushright y86
  itemize A232
  lrbox *232*
  math z193
  minipage *233*
  sloppypar J48
  thebibliography *307*
  verbatim* y121
\epsilon t191
\eqnarray z210, z257
eqnarray (environment) z205, z305
eqnarray* (environment) z256
\eqno z199
\equation z198
equation (environment) z197, z293
\equiv t356
\err@rel@i q89, q122, q126
\errhelp a164, c23, g42, g69, M12, N80, N234
\errmessage a4, a169, b74, c24, g50, g75, o389, p385, p515, q55, M16, N82
\error@fontshape o370, o390, p107, p517, r155

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\errorcontextlines b94, b126, b252, g181
\errorstopmode b244, N242
\escapechar d95, d140, d145, d154, o286, o420, p183, r54, r113, r154
\eta t193
\evensidemargin K54, K492
\every@math@size o51, p189, p201
\everycr b222, z135, z138, z218, z320
\everydisplay o262, o263, o268
\everyjob c28, r174, N204, N205, N207
\everymath o261, o263, o266
\everypar k37, o477, *46*, y50, y116, A126, A127, A170, A187, B183, C72, F31, F79, F90, F110, F119, G112, K142, K890
\execute@size@function p318, p346, p364, p381
\ExecuteOptions l904, p57, p70, L194
\exhyphenpenalty b108, b195
\exists t249
\exp z31
\external@font p84, p87, p98, p102, p104, p347, p365, p439, p477, p549, p551, p553
\extra@def q1, q74
\extracolsep C149
\extract@alph@from@version o426, o432, r117
\extract@font o297, p81
\extract@fontinfo p314, p321
\extract@rangefontinfo o314, p331, p338, p361, p394
\extract@sizefn o311, p305, p328

F

\f@baselineskip o234, o241, o382, p119, p136, p140, p155, p169, p180, p194
\f@encoding l133, o199, o218, o221, o222, o224, o243, o275, o280, o333, o335, o337, o342, o344, o374, o386, p91, p261, p507, r140
\f@family l921, l924, l938, l948, l954, l1109, o227, o237, o276, o280, o333, o335, o337, o342, o344, o378, o397, p91, r140
\f@linespread o237, p118, p137, p138, p141, p149, p152, p163, p166
\f@series j14, o227, o238, o277, o280, o379, o394, s67
\f@shape o227, o239, o278, o280, o380, o392
\f@size l135, l956, o63, o234, o240, o279, o381, o411, o453, o454, o457, o458, p119, p121, p134, p154, p169, p172, p175, p180, p187, p194, p206, p209, p215, p221, p238, p239, p242, p247, p315, p322, p341, p343, p362, p425, p427, p429, p445, p446, p451, p469, p485, p494, p512, p520, p525, p537, p555
\f@user@size p445, p450, p512, p525
\fam o24
\familydefault r171, s81, t38
\fbox *232*, B98, B107
\fboxrule B96, B118, B121, B127, B129, B136, B137, N8
\fboxsep B96, B102, B117, B122, B132, B134, N7
\filbreak b199
\filec@ntents L430, L431, L432, L511, L517
\filecontents L425
filecontents (environment) *362*
\filename@area a193, a199, a206, a212, a219, a225, a232, k168, k193, k196, k212, k224, k226
\filename@base a241, k168, k193, k196, k219, k224
\filename@dot a239, a242
\filename@ext a237, a239, k169, k189, k190, k193, k196, k220
\filename@parse *1*, *4*, a57, a189, k166, k188, k217
\filename@path a194, a195, a200, a207, a208, a213, a220, a221, a226
\filename@simple a197, a210, a223, a233, a235
\fill i195
\finalhyphendemerits b117
\finph@nt z87, z89, z90
\finsm@sh z103, z105, z106
\firstmark J37, K526
\fix@penalty v84
\fixed@sfcnt o326, p543, p544, p547
\flat t252
\floatingpenalty G275
\floatpagefraction G176, K1520
\floatsep K544, K562, K569, K1400, K1449, K1525
\flushbottom J41

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\flushleft y80
flushleft (environment) y80
\flushright y86
flushright (environment) y86
\fmtname c1, c28, c31, L243, L247
\fmtversion c1, c11, c30, c33, g1, o1, C1, D1, K4, L260, L263, N188, N214
\fmtversion@topatch N186, N188, N200, N201, N213, N221, O5
\fnsymbol m39, *103*
\font b227, b232, l251, l252, l253, l338, l345, l643, l650, o54, o62, o64, p84, s31, s54, s66, u8, u9, u10, v68, y115
\font@info p99, p321, p394, p401
\font@name l134, l137, o61, o169, o171, o271, o288, o410, p84, p88, p90, p105, p120, p123, p126, p284, p285, p286, p287, p288, p293
\font@submax p407, p442, p443, y22, y24, N72, N74, N83
\fontdimen b227, b232, l251, l252, l253, l338, l345, l643, l650, s31, s66, v68, D48, D50, D334
\fontencoding l871, o199, o230, r170, t14
\fontfamily l940, o227, r171, s5, s8, s11
\fontname o64
\fontseries o227, r172, s14, s17
\fontshape l348, l653, o227, r173, s20, s23, s26, s29
\fontsize j6, l256, l958, o52, o235, s60, G260
\fontsubfuzz p407, p447, y22
\footins G245, G271, K264, K265, K266, K267, K299, K358, K366, K370, K399
\footnote G262
\footnotemark F9, G284
\footnoterule B228, G249, K369
\footnotesep B247, G261, G274, G282
\footnotesize B240, G272
\footnotetext F11, G301
\footskip K58, K516
\forall t248
\fps@dbl G34
\frac z202
\frame B82, B144
\framebox *232*, B105
\frenchspacing b152, k40, y118, y144
\frown t359
\frozen@everydisplay o261, o267
\frozen@everymath o261, o265
\fussy J50
\futurelet d257, d271, i172, i180, v66, z153, C327

G

\g@addto@macro L369, L375, L379, L380
\G@refundefinedfalse x5
\G@refundefinedtrue I21, I44, x3, x12
\Gamma t216
\gamma t189
\gcd z33
\ge t330
\gen@sfcnt o322, p480, p481, p484
\genb@sfcnt o323, p489, p490, p493
\genb@x p494, p497
\genb@y p497
\GenericError g19, g91, g122, g153, p62
\GenericInfo g4, g112, g143, g173, p31, p34, p39, p75
\GenericWarning g11, g102, g133, g164, p42, p47, p50, p78
\geq t329, t330
\get@cdp r284, r292, r325
\get@external@font p83, p96, p526
\getanddefine@fonts o421, o439, p274, r55, r102, r114, r196, r255, r289, r291, r308, r431, r432, r464, r465, r811, r812
\GetFileInfo t3
\getlinechar D129
\gets t348
\gg t343
\glb@currsize k35, o258, p171, p206, p210, p216, p239
\glb@settings o259, p171, p218, p249
\globaldefs o422, p185, r56, r115
\glossary *305*, F146, H23, H35, J20, J28, K500
\glossaryentry H32
\goodbreak b199
\grave t425
\group@elt r35, r194, r226, r227, r248, r252, r843
\group@list r198, r233, r246, r251, r252, r281, r503, r545, r625, r628, r678, r681, r728, r731, r798, r849
\guillemotleft l403, l620
\guillemotright l404, l621
\guilsinglleft l405
\guilsinglright l406

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

H

\H g27, l185, l287, l365, l458, l466, l485, l493, l600
\h@false z77
\h@true z78, z79
\halign . . . b222, z96, z140, z218, z320
\hangindent F122
\hat t431
\hb@xt@ b237, d16, l324, z140, z223, z269, z283, z298, z328, B28, B43, B116, B289, B293, B294, C39, D21, D31, D40, D143, D177, D180, D183, D185, D187, D262, D291, D390, F163, F166, K509, K519, K1181, K1479, K1480, K1485
\hbadness b104, o480
\hbar t228
\headheight K56, K505
\headsep K57, K514
\heartsuit t257
\height B15, B18
\hexnumber@ r518, r526, r541, r560, r568, r576, r585, r588, r597, r598, r637, r645, r690, r698, r712, r713, r716, r741, r749, r754, r756, s71
\hfuzz b127, J46, J47, J53, J54
\hgl@ b192, b193
\hglue b189
\hideskip b79, b213
\hidewidth b213, l292, l293, l296, l299, l372, l373, l377, l380, l382, l384, l607, l608, l611, l614, l678, l681
\hline C326, C329
\hmode@bgroup l72, l78, l292, l298, l326, l337, l344, l372, l379, l382, l384, l577, l607, l613, l642, l649, l677, l680, l726, v7
\hmode@start@before@group l73, l116, l118, l124, l139
\holdinginserts b95
\hom z29
\hookleftarrow t388
\hookrightarrow t386
\hphantom z75
\hrule b190, b234, i148, i156, l245, l248, t246, t522, B88, B93, B127, B137, C327, C344, D293, G250
\hrulefill b234
\hspace i191
\hyphenation l160
\hyphenchar y115
\hyphenpenalty b107

I

\I b158, L485, L503, N49, N154
\i l202, l304, l351, l352, l353, l354, l355, l356, l407, l444, l445, l537, l539, l541, l543, l622, N54, N159, N166
\ialign b222, b224, t243, t367, t438, t441, t444, t447, z109, z111, z119, C173, D72
\if@afterindent F107, F114
\if@compatibility L2, L202
\if@endpe y62, A128
\if@eqnsw z205, z254
\if@fcolmade K76, K212, K318, K327, K597, K614, K641, K695, K1463, K1500
\if@filesw I4, I8, I19, I28, I36, I43, k7, k30, k92, k104, k111, k120, y14, y28, F136
\if@firstamp C221
\if@firstcolumn K76, K194, K227, K320, K1153, K1470
\if@ignore y4, y63
\if@inlabel A28, A65, A102, A150, A173, K138
\if@insert K76, K799, K908, K942, K1012, K1128
\if@minipage i102, i115, y101, A139, B191, C70, G20
\if@mparswitch K76, K1155
\if@multiplelabels x31
\if@negarg D76, D98, D112, D151
\if@newlist y119, A29, A33, A69, A78, A106, A156, K476, K523
\if@nmbrlist A33, A191
\if@no@font@opt q100, q119
\if@nobreak i58, i117, k67, k79, A157, A182, B177, F30, F111, G105, G228, J25, J33, K142, K287, K887
\if@noitemarg A32, A189
\if@noparitem A30, A147
\if@noparlist A31, A114
\if@noskipsec A58, B178, F21, F23, F80, G229, K132
\if@ovb D233, D279, D284
\if@ovl D233, D277, D294
\if@ovr D233, D276, D292
\if@ovt D233, D278, D288

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

`\if@partsw` k7, k96
`\if@pboxsw` B174, B249
`\if@reversemargin` K82, K1158
`\if@reversemarginpar` K76
`\if@rjfield` C21, C35
`\if@specialpage` K76, K485
`\if@tempswa` I52, a25, a26, a27, e9, k102, o74, r219, r269, r333, r414, r842, y30, y107, K732, K768, K1093, L448
`\if@test` K12, K13, K677, K710, K774, K855, K864, K993, K1098
`\if@twocolumn` k20, G32, G131, K76, K120, K215, K226, K319, K343, K599, K632, K1152, K1464
`\if@twoside` K76, K119, K488
`\ifdt@p` z133, z135
`\iff` t408
`\IfFileExists` a125, k134, k161, k172, *60*, *365*, N182
`\ifG@refundefined` x3, x4, x5
`\ifh@` z76, z93
`\ifin@` l861, l864, q40, q42, r1, r22, r183, r280, r282, r343, r356, r426, r428, r456, r504, r515, r546, r557, r626, r629, r649, r679, r682, r726, r729, r732, r799, r801, r830, L80, L155, L167
`\ifinner` z169, z183, G53, G194
`\ifmath@fonts` o179, p176
`\ifmaybe@ic` v65, v74
`\ifnot@nil` p297, p316, p337
`\ifodd` r773, D192, D212, G64, K21, K119, K489, K724, K727, K760, K763, K871, K874, K1041, K1044, K1156, K1335, K1343
`\iftc@forced` l896, l906, l1115
`\ifv@` z75, z92
`\ifvbox` K270, K300, K379, K400
`\ignorespaces` I7, I9, i24, i74, i83, i207, k60, o232, y63, y71, y72, z191, A55, A207, B79, B247, C59, C60, C61, C74, C83, C96, C100, C107, C114, C116, C125, C207, C269, C271, C273, C300, D24, D32, D43, D74, D75, E30, E32, F93, G17, G24, G282
`\ignorespacesafterend` y7
`\IJ` l205, l335, l410
`\ij` l204, l333, l409
`\Im` t234
`\imath` t229
`\in` t340, t369
`\in@` l859, l862, q39, q41, r1, r21, r182, r279, r281, r339, r352, r425, r427, r454, r502, r513, r544, r555, r624, r627, r647, r677, r680, r724, r727, r730, r797, r800, r828, L79, L152, L166
`\in@@` r5, r6, r7, r9
`\in@false` r10
`\in@true` r12
`\include` k86, *60*
`\includeonly` k82, *60*
`\indent` A151, C72
`\index` *305*, F146, H6, H18, J20, J28, K499
`\indexentry` H15
`\inf` z25
`\infty` t236
`\init@restore@glb@settings` p219, p222, p224
`\init@restore@version` r58, r73, r93, r94
`\input` a15, a121, a124, a181, d7, k163, l1094, p16, q8, q96, s131, s142, s152, t10, t11, t12, t13, t20, t21, t25, t26, t55, t56, t57, t58, t540, t541, t542, *60*, *365*, L206, N69, N187
`\input@path` *1*, *4*, a56, a78, a80, a86, a88, a94, a96, a101, a103, a113, a180, k137, k151
`\InputIfFileExists` k160, k165, k173, k192, l849, l1176, o342, s105, s123, s134, s144, *60*, *365*, L332, M8, N63
`\inputlineno` a250, b86, b87, b88, g183, g186, s104, N34, N45, N53, N128, N139, N150, N158
`\insc@unt` b37, b47, b48, b49, b51, b62, b63, b64, b65, b66, b67
`\insert` b69, G271, K399, K400, K1230
`\install@mathalphabet` o416, o433, o440, r202, r205, r286, r287, r384, r436, r439, r446, r461, r462, r469, r813, r815
`\int` t266
`\interdisplaylinepenalty` i29, z55, z137, z240
`\interfootlinepenalty` b148
`\interfootnotelinepenalty` b148, i34, G273
`\interlinepenalty` i27, y108, y111, F50, F101, F154, G273, K290, K892, K896

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\intextsep K875, K879, K894, K897, K904, K1525
\intop t265, t266
\iota t195
\is@range o315, p332, p333
\ishortstack D62
\itdefault s29, t34
\item I4, I8, g277, y73, y80, y86, y100, z282, z297, A131, A209, C69, E36, E38
\itemindent A9, A42, A95, A177, A198
\itemize A232
itemize (environment) A232
\itemsep A1, A166
\iterate a28, a29, b178
\itshape l346, l651, s27, s28, s32, v21, E36, E38, G254

J

\J N51, N156
\j l203, l305, l408, l623, N166
\jmath t230
\Join s91
\joinrel t379, t386, t388, t390, t392, t394, t396, t398, t400, t404, t406
\jot z53, z134, z247

K

\k l381, l447, l452, l474, l479, l555, l556, l604, l605, l656, l658, l663, l665
\kappa t196
\ker z27
\kernel@ifnextchar d48, d68, d119, d258, d275, L112
\kill C61

L

\L l197, l323, l393, l618, L482, L502, N179
\l l206, l325, l411, l624, N179
\l@ngrel@x d41, d42, d43, d89, d138
\label x32, F146, J20, J28, K498
\labelsep A9, A200, A206, E36, E38
\labelwidth A9, A93, A199, A201, A204
\Lambda t219
\lambda t197
\land t283
\langle t498
\language b35, b56, b89, b90, M10
\last@fontshape o388, o403
\lastbox z123, z124, A127, A175, F82, F115, K255
\LastDeclaredEncoding o112, o115
\lastpenalty v95, v98
\lastskip b202, b203, b205, b207, i19, i66, i87, i88, i92, i94, i95, i103, i119, i122, i123, v85, v88, A115, A116, A140, A141, D44
\LaTeX j3, j15, L451
\LaTeXe j13
\lbrace l262, t502
\lbrack b162
\lccode g22, g23, g24, g25, g26, g27, l109, y139, y149, N31, N41, N50, N52, N54, N56, N59, N60, N61, N62, N136, N146, N155, N157, N159, N161, N164
\lceil t506
\ldotp t409, t412, t523
\ldots l276, t413
\le t328
\leaders b234, t246, t464, t465, t467, t468, C344, D287, D293, F159
\leadsto s94
\leavevmode I14, b193, b220, b223, b234, b236, i169, i183, l78, l139, l243, l245, l295, l324, l328, l331, l375, l610, l640, l952, v106, y108, y119, y132, y150, z282, z297, A58, A103, B4, B8, B81, B83, B99, B113, B159, B207, B252, B265, C160, D65, D187, F23, F155, G296, K134, K139
\left t524, t525, t526, t527, z108, z114, z125
\Leftarrow t324, t400, t406
\leftarrow t347, t348, t388, t398, t404, t456
\leftarrowfill t442, t456
\lefteqn z259
\leftharpoondown t361, t375
\leftharpoonup t360
\lefthyphenmin b92, M11
\leftline B289
\leftmargin A9, A52, A53, A94, A136, A138
\leftmargini z275, A17
\leftmarginii A17
\leftmarginiii A17
\leftmarginiv A17
\leftmarginv A17
\leftmarginvi A17
\leftmark J34
\Leftrightarrow t323
\leftrightarrow t346

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

`\leftskip` b215, y77, y84, y90, y102, A74, B186, F152, F157
`\leq` t327, t328
`\lfloor` t510
`\lg` z4
`\lgroup` t512
`\lhd` s97
`\lhook` t385, t386
`\lim` z6
`\liminf` z8
`\limits` t446, t450, z107, z201
`\limsup` z7
`\line` g262, D77, D260
`\linebreak` i13, *48*
`\linepenalty` b106
`\lineskip` b156, b188, b223, t366, z130, B187, C62, C180, D67, D188, K501
`\lineskiplimit` b157, b188, b225, b226, t366, t418, z132, z136, K501
`\linespread` o233
`\linethickness` D56
`\linewidth` k24, z176, z283, z298, z302, z320, A15, A51, A52, A54, B184, C38, G164, K127, K153
`\list` A34, A226, A237
`\listfiles` k206, *365*
`\listparindent` A9, A41, A50
`\ll` t344
`\llap` A228, A239, B293, B294
`\lmoustache` t469
`\ln` z5
`\lnot` t251
`\LoadClass` *363*, L212, L226, L349, L408, L416, L417
`\LoadClassWithOptions` *364*, L225
`\log` z3
`\loggingall` b248
`\loggingoutput` b241, b252
`\Longleftarrow` t400
`\longleftarrow` t397
`\Longleftrightarrow` t406, t408
`\longleftrightarrow` t404
`\longmapsto` t402
`\Longrightarrow` t394
`\longrightarrow` t395, t402
`\loop` a28, b178, C350
`\lor` t285
`\lower` j2, t366, B125, D23, D96, D183, D184, D221, D222
`\lower@bound` p342, p343, p355
`\lowercase` g29, l110, l847, o249, o341, y143, y150, N175
`\lq` b160
`\lrbox` B69
`lrbox` (environment) *232*
`\ltx@sh@ft` b229, l292, l299, l372, l380, l607, l614

M

`\M` b158
`\m@ne` b39
`\m@th` b209, b221, j13, t243, t367, t369, t370, t373, t414, t438, t441, t444, t447, t453, t456, t463, t466, t528, z68, z71, z89, z105, z108, z110, z115, z134, z214, z283, z298, z308, B174, B263, C163, F159, G255, G260
`\magstep` b149
`\magstephalf` b149
`\makeatletter` d272, d288, k26, o305, o347, y19, F134, K2, L206, L311
`\makeatother` d272, L206, N241
`\makebox` z176, *232*, B3
`\makeglossary` k69, *305*, H20
`\makeindex` k68, *305*, H3
`\makelabel` A45, A97, A195, A208, A228, A239
`\MakeLowercase` N172, N181
`\makeph@nt` z84, z86
`\makesm@sh` z100, z102
`\maketitle` *282*
`\MakeUppercase` N165, N165
`\mandatory@arg` p374, p477, p485, p494, p507, p509, p514, p516, p521, p523, p538, p549, p551
`\mapsto` t352
`\mapstochar` t351, t352, t402
`\marginpar` G187
`\marginparpush` K66, K1172
`\marginparsep` K65, K1183, K1185
`\marginparwidth` G216, K64, K1185
`\mark` J23, J31, J39
`\markboth` J18
`\markright` J18
`\math` z193
`math` (environment) z193
`\math@bgroup` o447, p260, p266, r49, r108, v113, v114, v121
`\math@egroup` o447, p264, p265, v114, v115, v122
`\math@fonts` o417, o422, p186, p290, r56, r115

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\math@fontsfalse j7, l256, l957, o50, o181, o189
\math@fontstrue o179, o461
\math@version o10, o253, o421, o425, o427, o428, o430, p184, r52, r55, r60, r61, r65, r76, r77, r78, r96, r97, r98, r111, r114, r118, r120, r122, r126, s53
\mathaccent r513, r541
\mathalpha r612, r771, 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 s13, t70
\mathbin r776, 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 b221, r555, r597, t228, t240, t521
\mathchar@type r541, r585, r588, r597, r613, r712, r772
\mathchardef b21, b22, b23, b24, e3, e4, e5, e6, l75, r588
\mathchoice z61
\mathclose r779, t150, t159, t161, t164, t169, t175, t177, t179, t472, t497, t501, t505, t509, t515, z43, z46, z49, z52
\mathcode r585, t171, t172, t173
\mathdollar l261, t518
\mathellipsis l275, t523
\mathgroup l1104, o22, p257, p263, p269, p270, p281, s68, t529
\mathhexbox b221, s78
\mathindent z273, z284, z300, z310
\mathinner t412, t416, t421, t523
\mathit s28, t72, t75, t521
\mathnormal t68
\mathop r775, 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, z201
\mathopen r778, t160, t163, t168, t174, t176, t178, t470, t499, t503, t507, t511, t513, z41, z44, z47, z50
\mathord r612, r774, 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 l264, m59, t518
\mathph@nt z82, z88
\mathpunct r780, t153, t157, t409, t410, t411
\mathrel r777, 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, z201
\mathring t436
\mathrm s4, t67
\mathsection l265, m59, t518
\mathsf s7, t71, t74
\mathsm@sh z98, z104
\mathsterling l273, t518
\mathstrut z94, z112, z113
\mathsurround b209
\mathsymbol r590
\mathtt s10, t73
\mathunderscore t518
\mathversion o253, s50, s52
\matrix z110, z114
\max z22
\maxdeadcycles K7
\maxdepth b130, k50, K73, K384, K394, K428, K533, K542, K582, K733, N9
\maxdimen b79, b131, b132, b188, b226, b244, b252, o478, p340, p399, t366, D264, D298, K239, K1193, K1220, K1225, N13
\maybe@ic v46, v47, v66
\maybe@ic@ v66
\maybe@icfalse v80
\maybe@ictrue v70
\mb@b B34, B44
\mb@l B34, B38, B43, D68, D72
\mb@r B34, B38, B43, D68, D72
\mb@t B35, B42
\mbox b221, j13, l247, s74, t414, *232*, B7, B8, D28, G260
\mddefault s17, t32, t40
\mdseries s15, s16, s77, v20
\meaning a166, a175, a246, d201, d281, r340, r353, r454, r513, r555, r647, r724, r828
\medbreak b203
\medmuskip t531, z36, z38, z145
\medskip b206, i162
\medskipamount b205, i163, i165
\MessageBreak d177, g3, g6, g13, g36, g49, g63, g76, g197, g199, g205, g217, l126, l852, l855, l879, l880, l881, l882, l883, l884, l885, l886, l887, l888, l937, l939, l947, l954, l1109, o401, p20, p21, p67, p88, p281, p450, p474, p520, p537, p554, q21, q23, r295, r304, r442, v127, y23, K460, K1293, L87, L236, L247, L249, L251, L262, L318, L319, L321, L322, L323, L325, L327, L344, L345, L346, L347, L393, L410, L411, L443, L471, N73, N74, N75, N77
\mho s90
\mid t316
\min z23
\minipage B194
minipage (environment) *233*
\mit s155
\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
\month a132, c9, L455
\moveright K504
\mp t303
\mscount C347
\mskip i188, z36, z38, z144, z145, z146, z147
\mu t198
\multicolumn C203
\multiput D33, D37
\multispan C203, C347
\muskip b29, b50, t451, t452
\muskipdef b50

N

\n@space . . t524, t525, t526, t527, t528
\nabla t239
\narrower b214
\natural t253
\ne t326
\nearrow t319
\NeedsTeXFormat p12, L241, L514
\neg t250, t251
\negthinspace i198
\neq t326
\new@command d44, d45, d100, d135, d157, d213
\new@environment . . . d115, d116, d129
\new@fontshape q1, q14
\new@mathalphabet . . . r337, r358, r369
\new@mathgroup o22, r221
\new@mathversion r20, r179, r181
\new@symbolfont r222, r250
\newbox b47, b84, b211, e13, z66, A27, B54, C16, C17, C18, C311, D6, D325, D330, K67, K101, K102, K103

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\newcommand . . . *23*, d44, l4, t29, t30, t31, t32, t33, t34, t35, t36, t37, t38, t39, t40, t41, D341, K1513, K1516, K1519, K1520, K1523, K1524
\newcount . . . b47, b87, b90, b92, b93, b94, b95, b97, b99, b148, e7, e8, i62, k9, m25, p25, r27, r187, z55, z205, z206, A23, A24, A25, A26, A56, A216, A231, B235, C11, C12, C13, C14, C15, C303, C304, C305, D319, D320, D321, D322, D331, F19, F123, F124, G3, G165, G166, G167, G168, K84, K86, K88, K90, K92, K100, K1282, K1511, K1514, K1517, K1521, N3, N4, N5
\newcounter . . . m10, *103*
\newdimen . . . b47, b79, b81, b82, b96, b147, e10, e11, e12, i61, p356, p357, z53, z274, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20, A21, A22, B96, B97, C3, C5, C6, C7, C8, C143, C306, C307, C308, C309, D3, D4, D5, D7, D239, D240, D241, D242, D243, D244, D323, D324, D326, D327, D328, D329, G261, K52, K53, K54, K56, K57, K58, K59, K60, K61, K62, K63, K64, K65, K66, K72, K74, K75, K87, K89, K91, K93, K94, K95, K96, K97, K98, K99, K1283, K1284
\newenvironment . . . *23*, d115, L453
\newfam . . . o25
\newfont . . . s54
\newgroup . . . r47
\newhelp . . . b47
\newif . . . d139, e9, k7, k8, l896, o179, r15, v65, x3, z75, z76, z133, z207, A28, A29, A30, A31, A32, A33, A128, B249, C21, C221, D76, D233, D234, D235, D236, F21, F107, K76, K77, K78, K79, K80, K81, K82, K83, L2
\newinsert . . . b62, B236, G245, K23, K24, K25, K26, K27, K28, K29, K30, K31, K32, K33, K34, K35, K36, K37, K38, K39, K40, K1192
\newlabel . . . x22, x34
\newlanguage . . . b56
\newlength . . . n3, *106*
\newline . . . i43
\newlinechar . . . a19, d5
\newmathalphabet . . . q99
\newmathalphabet@@ . . . q99
\newmathalphabet@@@ . . . q99
\newmuskip . . . b47
\newpage . . . K114, K120, K131
\newread . . . b54, k3
\newsavebox . . . *232*, B54
\newskip . . . b47, b80, b83, b145, b146, e14, e15, e17, i165, i166, i167, i195, n3, y79, z208, A2, A3, A4, A5, A6, A7, A8, K1525, K1526, K1527, K1531, K1532, K1535, K1536, K1537, K1541, K1542, K1543
\newtheorem . . . E1
\newtie . . . l695, l1036
\newtoks . . . b47, e16, o263, o264, p201
\newwrite b54, k4, k5, k6, F137, H4, H21
\nfss@catcodes . . . d290, o28, o95, o305, o338, o339, o346, q10, t19, t24, t54, K3
\nfss@text l269, l271, s74, v5, v105, x13
\NG . . . l394, N179
\ng . . . l412, N179
\ni . . . t341, t342
\no@alphabet@error . o6, r201, r203, r374, r375, r389, r398, r484, r485
\noaccents@ . . . o469, t48
\noalign . . . t245, t439, t442, t444, t445, t449, t450, z112, z113, z118, z121, z135, z247, C202, C327, C346, D75
\noboundary . . . b98
\nobreak . . . b191, b194, b196, i38, i53, i149, i157, i176, i183, i193, k67, k79, l334, l336, y69, B288, F73, F157, F158, F162, G297, J25, J33, K288, K888
\nobreakdashes . . . i168
\nobreakspace . . . i182
\nocite . . . I39, *307*
\nocorr . . . v26, v41, v45, v48
\nocorrlist . . . v72, v104
\nofiles . . . k63, *60*
\noindent . . . F122
\nointerlineskip . . . b186, t245, t439, t442, t445, t449, z175, D285, D287, K1180, K1188
\nolimits . . . t266, t273, z3, z4, z5, z9, z10, z11, z12, z13,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

z14, z15, z16, z17, z18, z19, z20, z21, z26, z27, z28, z29, z31, z34
\nolinebreak i13, *48*
\non@alpherr o441, o443, r68, r82, r129, r850
\nonfrenchspacing . . . b152, b257, k42
\nonscript z36, z38
\nonumber z234, z257, z258
\nopagebreak i3, *48*
\normalbaselines . . . b156, z108, z110
\normalbaselineskip b145, b157, p142, B188
\normalcolor z200, z270, B47, B227, F163, G93, K164, K368, K508, K518, K1483
\normalfont o479, s79, v18, y120, z200, z270, F163, G256
\normallineskip . . . b145, b156, B187
\normallineskiplimit b145, b157, z136
\normalmarginpar G242
\normalsfcodes k38, k40, k42, k62, K497
\normalsize k36, v125, G23, G101, G227, K496, L5
\not t244, t326, t345
\not@base s86, s90, s91, s92, s93, s94, s95, s96, s97, s98, s99, s100
\not@math@alphabet s4, s7, s10, s13, s16, s19, s22, s25, s28, s33
\notin t369
\notrace K1244
\nu t199
\null b170, l382, l384, l678, l681, x17, y108, y132, z91, z110, z128, F157
\nulldelimiterspace b134, t528
\nullfont y51
\number a33, d2, d82, m40, o425, o428, p401, r60, r78, r98, r119, s71, L424, L455
\numberline F55, F65, F166, G17
\nwarrow t321

O

\O l199, l302, l396, l617, N178
\o l208, l307, l414, l625, N178
\o@lign b223, l292, l299, l372, l380, l607, l614
\oalign b223
\obeycr i204
\obeylines b173, y114, y127, y128, K465
\obeyspaces b173, K465
\oddsidemargin K53, K55, K490
\odot t298
\OE l198, l301, l395, l616, N178
\oe l207, l306, l413, l626, N178
\of z67, z204
\offinterlineskip b186
\oint t273
\ointop t272, t273
\oldstylenums l1101, s64
\Omega t226
\omega t209
\ominus t301
\omit z121, z122, C337, C340, C347, C351
\on@line g8, g15, g183, y56, B75, L343
\onecolumn K122
\OnlyDescription p5, u3
\ooalign b223, l296, l376, l382, l384, l578, l611, l678, l681, l727, s76, 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 p375, p470, p472, p548, p551
\OptionNotUsed L136, L143, L358
\Orb l561
\oslash t299
\otimes t300
\outer b11
\outer@nobreak G106, G145, G150, G153, G221
\outerparskip A1
\output K204
\outputpenalty K206, K220, K245, K248, K249, K286, K898, K899
\oval D260, D263
\over t377, z107, z202
\overbrace t444
\overfullrule b129, J55
\overleftarrow t441
\overrightarrow t438
\owns t342

P

\P l264
\p@ b81
\p@equation z212, z307
\p@reset@font s79
\p@selectfont p117
\PackageError g88, l850, l902, l946

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\PackageInfo . g88, l879, l892, l893, l953, l1177
\PackageWarning g88, l903, l1107
\PackageWarningNoLine g88
\pagebreak i3, *48*
\pagegoal K1227, K1234
\pagenumbering w5, *198*
\pageref x10
\pageshrink K420, K424, K440
\pagestyle J2
\pagetotal K109
\paperheight K74
\paperwidth K74
\par a67, b11, b166, b174, b175, b190, b199, b200, b201, b203, b205, b207, d6, h3, h4, h6, y49, y69, y106, A63, A110, A126, A151, A154, B179, B223, C177, C353, F24, F73, F164, G15, G24, G143, J48, J49, K143, K205, K1233
\par@deathcycles A56, A77, A79, A80
\paragraphmark F126
\parallel t315
\parbox *232*, B145
\parboxrestore B190
\parfillskip b144, o478, y78, y91, y103, A76, B187, F152
\parindent . b136, b215, b216, y78, y85, y91, y103, A50, B182, F153
\parsep A1, A49, A90
\parshape A54
\parskip b137, y70, y101, y103, z316, A49, A73, A88, A90, A117, A143, A162, A213, B182, C70, K898
\partial t235
\partopsep z314, A1, A61
\PassOptionsToClass *364*, L113
\PassOptionsToPackage *363*, L113
\patch@level N189, N201, N203, O8
\patterns l160
\penalty I17, b195, b196, b197, b198, b199, b200, b204, b206, b208, i7, i10, i21, i120, i124, v101, y108, y111, z37, z137, z247, A180, C58, G120, G124, G126, K117, K145, K146, K896
\perp t355
\ph@nt z77, z78, z79, z80
\phantom z75
\Phi t224
\phi t206
\Pi t221
\pi t201
\pickup@font l136, o170, o270, o412, p122, p285, p287, p289
\pictur@ D9, D53, D54, D60
\picture D9
\pm t304
\pmatrix z114
\pmod z39
\poptabs g244, C129
\poptracing p130, p294
\postdisplaypenalty i28, z281, z295
\pounds l272
\Pr z32
\pr@@@s z156, z164
\pr@@@t z159, z165
\pr@m@s z153, z154
\prec t332
\preceq t335
\predisplaypenalty b115, z280, z294
\preload@sizes q84
\pretolerance b102, o480
\prevdepth b186, b190, b191, i147, i152, z135, G121, G123
\prim@s z150, z152, z164
\prime t172, t237, z153
\prime@s z151
\process@table k34, r133
\ProcessOptions l870, l905, p71, L144, L187, L412
\ProcessOptions* L144
\prod t267
\propto t312
\protect I5, d70, d192, d208, d217, d222, d225, d226, d228, d229, d234, d235, d240, d243, d244, g213, g232, g234, g235, g244, g252, g262, g274, g277, g285, k75, l26, l32, l53, l60, l164, l172, r403, r855, s57, v126, x12, C234, F11, F55, F65, F143, G17, K474
\protected@edef d227, x37, B243, F43, G277, N169, N175, N180, N181
\protected@write . . k66, k71, x33, F145, H14, H31
\protected@xdef . . . d227, F10, G263, G287, G303
\provide@command d151, d152
\providecommand d151, l6, K1266
\ProvidesClass *363*, L100

\ProvidesFile a36, t551, t553, t554, t555, L102
\ProvidesPackage p13, *363*, L83, L100, L515
\ProvideTextCommand l3, l65
\ProvideTextCommandDefault l62
\ps@empty J10, N15
\ps@plain J13
\Psi t225
\psi t208
\pushtabs g244, C126
\pushtracing p115, p275
\put D29, D197, D198, D199, D200, D205, D207, D219, D220, D221, D222, D227, D230, D378

Q

\qbezier *260*, D341
\qbeziermax D337, D362
\qquad i201
\quad i201, z109, z111, z120, F94
\quotedblbase l415, l627
\quotesinglbase l416

R

\r b164, b165, l191, l290, l330, l366, l467, l494, l504, l530, l603, l639
\r@@t z66
\radical r722, r724, r754
\raggedbottom J39
\raggedleft y86, y88
\raggedright y80, y82
\raise l329, l332, l579, l641, l728, s77, t373, t421, t423, z73, B273, B282, D30, D40, D95, D183, D262, D279, D305, D389
\raisebox *233*, B264
\rangle t496
\rbrace l263, t500
\rbrack b162
\rceil t504
\Re t233
\ref x10
\refstepcounter x32, z198, z296, *103*, A192, E27, F42, G9
\Relbar t384, t392, t394, t400
\relbar t381, t396, t398
\relpenalty b110
\rem@pt o246
\remove@angles p301, p326
\remove@nil r36
\remove@star p301, p309
\remove@to@nnil o245, p301, p329, p458
\removelastskip b202, b204, b206, b208
\renew@command . . d93, d94, d158, d166
\renew@environment d121, d122
\renewcommand *23*, d93, z269, z276, z286
\renewenvironment *23*, d121, z293, z305
\repeat a28, a30, b178, C350
\RequirePackage *363*, L202, L209, L230, L408
\RequirePackageWithOptions *364*, L228
\reserved@a a68, a72, a73, a142, a143, a146, a164, a168, a190, a197, a200, a202, a203, a210, a213, a215, a216, a223, a226, a228, a254, a255, a256, c5, c11, c26, d85, d89, d102, d103, d104, d106, d157, d158, d159, d165, d166, d167, d168, d171, d190, d199, d204, d255, d264, f33, f37, g216, i171, i174, k76, k77, k99, k100, k138, k140, k145, k147, k149, k155, k159, k167, k170, k186, k187, k191, k197, k218, k222, k226, l80, l82, l90, l107, l112, o38, o41, o44, o80, o83, o85, o122, o126, o340, o343, o387, o388, o400, o403, o408, o433, o436, o437, o445, p150, p152, p154, p164, p166, p169, p298, p299, p314, p315, q43, q47, r284, r293, r295, r339, r342, r352, r355, r453, r455, r513, r514, r555, r556, r647, r648, r724, r725, r827, r829, r845, r847, r848, r853, v30, v31, v36, v37, v48, v51, v71, v78, y41, y42, y54, y55, y59, y64, y65, z249, z250, z251, z252, z254, B36, B37, B40, B70, B76, C211, C215, C220, C239, C328, C329, D99, D101, D105, D266, G29, G30, G32, G33, G59, G63, G69, G72, G75, G78, K668, K1351, K1353, K1359, K1362, L77, L80, L81, L195, L198, L242, L243, L246, L283, L287, L299, L300, L302, L312, L352, L517, L519, N16, N33, N35, N36, N44, N46, N47, N85, N116, N122, N123, N125, N127, N131, N138, N140, N141, N149, N151, N152, N167, N168,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

N169, N170, N173, N174, N175, N176, N202, N205, N206, N223
\reserved@b a69, a70, d77, d79, d87, d104, d105, d200, d201, d204, d256, d266, f33, f34, f37, i172, i173, i180, k98, k100, k150, k152, k154, k221, k227, l83, l90, o70, o72, o125, o126, o434, o445, q37, q44, q61, q63, r215, r217, r265, r267, r292, r293, r294, r329, r331, r410, r412, r457, r458, r459, r466, v35, v36, v49, v51, v78, v79, C216, C218, C220, G39, G40, K604, K607, K620, K623, L78, L79, L291, L297, L300, L460, L461, L463, L489, N19, N21, N25, N88, N90, N94, N168, N174, N223
\reserved@c a70, a75, d261, d264, d266, d269, k210, k211, o71, o72, o435, o438, q38, q45, q51, q58, r33, r37, r216, r217, r266, r267, r330, r331, r411, r412, r434, r443, r458, r472, r637, r653, r662, r690, r701, r740, r753, r755, v50, v52, v59, L437, L438, L439, L449, L465, L472, L497, N23, N28, N38, N92, N113, N114, N115, N117, N118, N119, N120, N121, N124, N126, N133, N143, N225
\reserved@d a73, a76, d254, d263, k209, k211, q51, q58, q60, q64, r645, r653, r662, r698, r701, r748, r753, r757, N226
\reserved@e i36, i38, i47, i53, q29, q35, q60, q63, q64, r34, r39, N227
\reserved@f i37, i38, i53, l846, l847, l848, l849, l851, l858, o165, o167, o173, o174, p338, p349, p353, p361, p367, p370, p421, p458, p461, q17, q28, q35, q61, q63, N228
\reset@font I20, s79, x13, B240, G100, G226, G272, J14, K495
\restglb@settings p222, p232
\restore@mathversion r72, r75, r95, r103
\restore@protect d227
\restorecr i204
\reversemarginpar G242
\rfloor t508
\rgroup t512
\rhd s99
\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
\rightharpoondown t363
\rightharpoonup t362, t374
\righthyphenmin b93, M11
\rightleftharpoons t372
\rightline B289
\rightmargin A9, A40, A51
\rightmark J34
\rightskip b216, y77, y83, y90, y102, A75, B186, F152
\rlap l329, l332, l641, z259, z270, B293, C72
\rlh@ t372, t373
\rmdefault s5, s67, t29, t39
\rmfamily s3, s4, v15
\rmoustache t471
\Roman m36, *103*
\roman m35, *103*
\romannumeral m41, m42, A43, A224, A235
\root z66, z204
\rootbox z66
\rq b160
\rule *233*, B247, B250, G282

S

\S l265
\s@fct@ p386, p463
\s@fct@fixed p542
\s@fct@gen p479
\s@fct@genb p488
\s@fct@sgen p479
\s@fct@sgenb p488
\s@fct@sub p501
\s@fct@subf p530
\samepage i27, *48*
\savebox *232*, B55
\sb z142
\sbox b210, j4, *232*, A195, B57, B58, B63, B68
\scan@@fontshape q30, q33
\scan@fontshape q2, q3, q4, q5, q6, q7, q8, q16, q27

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\scdefault s26, t34
\scriptfont p292
\scriptfont@name p287, p292
\scriptscriptfont p293
\scriptscriptstyle z65, z68
\scriptspace b135
\scriptstyle t243, z64
\scshape l254, s24, s25, v23
\searrow t320
\sec z20
\secdef F125
\sectionmark F126
\select@group o418, o437, r48, r169, r206, r339, r392, r401, r439, r471
\selectfont j7, l256, l348, l653, l871, l940, l958, o231, p112, s5, s8, s11, s14, s17, s20, s23, s26, s29, s60, G258
\seriesdefault r172, s82, t38
\set@@mathdelimiter r699, r715
\set@color B46
\set@display@protect d3, d225, g7, g14, g37, g64
\set@fontsize . o234, o236, p119, p132
\set@mathaccent r516, r524, r540
\set@mathchar r574, r584
\set@mathdelimiter . . . r650, r659, r711
\set@mathradical r177, r750
\set@mathsymbol r558, r566, r587
\set@simple@size@args o313, p302, p317, p324, p345, p363
\set@size@funct@args o318, p307, p309, p371
\set@size@funct@args@ . . . o319, p371
\set@typeset@protect d225, d244, C179, C205, K480, K482
\setcounter k127, m2, m26, *103*, A215, K1512, K1515, K1518, K1522
\setlanguage b99
\setlength n4, z312, z317, z318, z319, *106*, B27, B115, B161, B164, B209, B254, B255, B256, B271, B272, B279, B280, B281, C158, C352, K1528, K1529, K1530, K1533, K1534, K1538, K1539, K1540, K1544, K1545, K1546
\SetMathAlphabet o14, q131, q132, r408, t74, t75
\SetMathAlphabet@ . . . r346, r415, r424
\setminus t307
\SetSymbolFont r263, t64, t65, t66
\SetSymbolFont@ r236, r270, r278
\settodepth n6, *106*
\settoheight n6, *106*
\settowidth n6, *106*
\sf@size j6, l256, o193, o459, p282, p286, G260
\sfcode b152, b153, b154, b155, b239, i178, k39, N42, N147
\sfdefault s8, t29
\sffamily s6, s7, v16
\sh@ft b227
\shapedefault r173, s83, t38
\sharp t254
\shipout K479
\shortstack D63
\showboxbreadth b124, b244
\showboxdepth b125, b244, o480
\showhyphens o472
\showoutput b241
\showoverfull b240, b245, b253
\Sigma t222
\sigma t203
\sim t353, t365
\simeq t354
\sin z9
\sinh z11
\sixt@@n a18, b16, b54, b55, o23, r52, r111, r508, r510, r550, r552, r593, r595, r633, r635, r641, r643, r686, r688, r694, r696, r736, r738, r744, r746, D156, D171, D173, G58, G80, K747, K793, K929, K1059, K1122, K1305, K1314, K1370, K1386, K1418
\size@update . . p128, p139, p158, p160
\sizefn@info p308, p310, p318, p346, p364
\skew t451
\skip b28, b49, b65, B226, G246, K266, K366
\skip@ b41, b189, b191, b192, b194, v88, v91
\skipdef b45, b49
\slash b195
\sldefault s23, t34
\sloppy B189, J43, J48
\sloppypar J48
sloppypar (environment) J48
\slshape l339, l644, s21, s22, v22
\smallbreak b203
\smallint t275

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\smallskip b204, i162
\smallskipamount b203, i162, i165
\smash t381, t453, t454, t457, t458, z95
\smile t358
\sp z142
\sp@n C347
\space b168
\spacefactor b193, b194, i67, i72, i80, i178, i190, l75, l76, G297, G299
\spaceskip s66
\spadesuit t258
\span C351
\split@name o274, o288, o371, p509, p523
\splitmaxdepth b131, G275
\splittopskip b143, G274
\sqcap t290
\sqcup t291
\sqrt z203
\sqrtsign t437, z71, z203
\sqsubset s95
\sqsubseteq t313
\sqsupset s96
\sqsupseteq t314
\SS l258, l397, N179
\ss l209, l308, l417, l628, N179
\ssf@size o194, o460, p282, p288
\stackrel z201
\star t311
\stepcounter m17, o430, r48, x36, z211, z254, z306, *103*, G262, G286, K525
\stop y49
\stretch i197
\strip@prefix a58, a175, a246, d201, d280, o415
\strip@pt b231, o188, o192, o246, o459, o460, p134
\strut b211, z121, z122, C31
\strutbox b211, p143, B247, C168, C169, G275, G282
\sub@sfcnt o324, p502, p503, p506
\subf@sfcnt o325, p531, p532, p535
\subparagraphmark F126
\subsectionmark F126
\subset t337
\subseteq t339
\subst@correction o60, o66
\subst@fontshape q1, q70
\subst@size p437
\subsubsectionmark F126
\succ t331
\succeq t334
\sum t268
\sup z24
\suppressfloats K1268
\supset t336
\supseteq t338
\surd t240
\sw@slant v74, v84
\swarrow t322
\symbol l127, s54
\symletters l1104, s68
\symoperators t529

T

\T g26, L498, L502, L503
\t l237, l588, l693, l884, l1092, l1093
\t@st@ic v73, v77
\tabbing C62, C145
\tabbingsep C121, C123, C142
\tabcolsep C229, C306
\tabskip b222, z138, z139, z216, z219, z222, z224, z310, z323, z326, z328, C149, C174
\tabular C156
\tabular* C157
\tabularnewline C176, C189
\tan z15
\tanh z17
\tau t204
\tc@check@accent l962, l1035, l1036, l1037
\tc@check@symbol l962, l1032, l1033, l1034, 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, l1071, l1072, l1073, l1074, l1075, l1076, l1077, l1078, l1079, l1080, l1081, l1082, l1083, l1084, l1085, l1086, l1087, l1088, l1089
\tc@error l943, l963
\tc@errorwarn l902, l903, l936
\tc@fake@euro l951, l1031
\tc@forcedfalse l896
\tc@forcedtrue l901
\tc@subst l935, l935, l962
\tencirc u10, D47, D333
\tencircw u10, D49
\tenln u9, D47, D48, D332, D334
\tenlnw u9, D49, D50
\TeX j1, j12

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\text@command . . . v8, v29
\textacutedbl . . . l748, l995
\textascendercompwordmark . l698, l978
\textasciiacute . . . l798, l1019
\textasciibreve . . . l746, l992
\textasciicaron . . . l747, l993
\textasciicircum . . . l241, l418
\textasciidieresis . . . l786, l1009
\textasciigrave . . . l737, l990
\textasciimacron . . . l793, l1014
\textasciitilde . . . l242, l419
\textasteriskcentered . . . l222, l564, l708, l985
\textbackslash . . . l223, l420, l565
\textbaht . . . l772, l1079
\textbar . . . l224, l421, l566
\textbardbl . . . l225, l567, l752, l998
\textbf . . . v19
\textbigcircle . . . l576, l725, l1055
\textblank . . . l705, l1040
\textborn . . . l738, l1058
\textbraceleft . . l226, l262, l422, l568
\textbraceright . l227, l263, l423, l569
\textbrokenbar . . . l784, l1007
\textbullet . . . l228, l570, l754, l1000
\textcapitalcompwordmark . . l697, l977
\textcelsius . . . l755, l1001
\textcent . . . l780, l1004
\textcentoldstyle . . . l757, l1066
\textcircled . . . l234, l238, l254, l255, l577, l726, l884, l1090, l1091
\textcircledP . . . l791, l1087
\textcolonmonetary . . . l759, l1067
\textcompsubstdefault . . . l940, l942
\textcompwordmark . . . l243, l424
\textcopyleft . . . l789, l1086
\textcopyright . l238, l271, l787, l1010
\textcurrency . . l782, l880, l883, l1034
\textdagger . l230, l266, l572, l750, l996
\textdaggerdbl l229, l267, l571, l751, l997
\textdblhyphen . . . l709, l1041
\textdblhyphenchar . . . l745, l1064
\textdegree . . . l794, l1015
\textdied . . . l740, l1060
\textdiscount . . . l774, l1081
\textdiv . . . l811, l1029
\textdivorced . . . l739, l1059
\textdollar . . . l210, l261, l337, l425, l642, l706, l983, l1096, l1098
\textdollaroldstyle . . . l756, l1065
\textdong . . . l768, l1076
\textdownarrow . . . l736, l1057
\texteightoldstyle . . . l719, l1050
\textellipsis . . . l250, l275
\textemdash . . . l211, l309, l426, l629
\textendash . . . l212, l310, l427, l630
\textestimated . . . l775, l883, l1033
\texteuro . . . l809, l881, l1030, l1031
\textexclamdown . . . l213, l311, l313, l428, l631
\textfiveoldstyle . . . l716, l1047
\textfloatsep . . . K546, K559, K1400, K1449, K1525
\textflorin . . . l758, l1002
\textfont . . . p291, z148
\textfont@name . . . p285, p291
\textfouroldstyle . . . l715, l1046
\textfraction . . K1260, K1392, K1519
\textfractionsolidus . . . l710, l986
\textgravedbl . . . l749, l994
\textgreater . . . l236, l429, l587
\textguarani . . . l762, l1070
\textheight . . . k16, k17, G155, G156, G159, G182, K59, K173, K174, K222, K323, K350, K524, K581, K612, N13, N14
\texthyphen . . . l215, l316, l431, l633
\texthyphenchar . l214, l315, l430, l632
\textinterrobang . . . l766, l1074
\textinterrobangdown . . . l767, l1075
\textit . . . v21
\textlangle . . . l721, l1052
\textlbrackdbl . . . l733, l988
\textleaf . . . l741, l1061
\textleftarrow . . . l703, l1038
\textless . . . l235, l432, l586
\textlira . . . l764, l1072
\textlnot . . . l790, l1012
\textlquill . . . l778, l1084
\textmarried . . . l742, l1062
\textmd . . . v19
\textmho . . . l724, l1054
\textminus . . . l722, l987
\textmu . . . l799, l1020
\textmusicalnote . . . l743, l1063
\textnaira . . . l761, l1069
\textnineoldstyle . . . l720, l1051
\textnormal . . . v15
\textnumero . . . l773, l1080
\textogonekcentered . . l383, l555, l556
\textohm . . . l732, l882, l1032
\textonehalf . . . l807, l1026
\textoneoldstyle . . . l712, l1043
\textonequarter . . . l806, l1025

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltdefns.dtx, e=ltalloc.dtx, f=ltcntrl.dtx, g=lterror.dtx, h=ltpar.dtx, i=ltspace.dtx, j=ltlogos.dtx, k=ltfiles.dtx, l=ltoutenc.dtx, m=ltcounts.dtx, n=ltlength.dtx, o=ltfssbas.dtx, p=ltfsstrc.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=ltfinal.dtx, O=ltpatch.ltx

\textonesuperior l803, l1023
\textopenbullet l776, l1082
\textordfeminine l259, l788, l1011
\textordmasculine l260, l804, l1024
\textparagraph l231, l264, l573, l800, l1021
\textperiodcentered l232, l574, l801, l1022
\textpertenthousand l387, l770, l1077, l1100
\textperthousand l385, l753, l999, l1099
\textpeso l763, l1071
\textpilcrow l771, l1078
\textpm l795, l1016
\textquestiondown l216, l312, l314, l433, l634
\textquotedbl l436
\textquotedblleft l217, l317, l434, l635
\textquotedblright l218, l318, l435, l636
\textquoteleft l219, l319, l437, l637
\textquoteright l220, l320, l438, l638
\textquotesingle l707, l984
\textquotestraightbase l699, l979
\textquotestraightdblbase l700, l980
\textrangle l723, l1053
\textrbrackdbl l734, l989
\textrecipe l765, l1073
\textreferencemark l802, l1088
\textregistered l254, l255, l792, l1013
\textrightarrow l704, l1039
\textrm v15
\textrquill l779, l1085
\textsc v21
\textsection l233, l265, l439, l575, l785, l1008
\textservicemark l777, l1083
\textsevenoldstyle l718, l1049
\textsf v15
\textsixoldstyle l717, l1048
\textsl v21
\textsterling l221, l273, l344, l440, l649, l781, l1005, l1095, l1097
\textstyle j15, t377, z63
\textsuperscript l257, l259, l260, G257
\textsurd l805, l1089
\TextSymbolUnavailable l3, l605
\textthreeoldstyle l714, l1045
\textthreequarters l808, l1027
\textthreequartersemdash l702, l982
\textthreesuperior l797, l1018
\texttildelow l744, l991
\texttimes l810, l1028
\texttrademark l257, l769, l1003
\texttt v15
\texttwelveudash l701, l981
\texttwooldstyle l713, l1044
\texttwosuperior l796, l1017
\textunderscore l244, l269, l441
\textup v21
\textuparrow l735, l1056
\textvisiblespace l246, l442
\textwidth k18, B214, G164, K60, K125, K149, K166, K509, K519, K1479, N14
\textwon l760, l1068
\textyen l783, l1006
\textzerooldstyle l711, l1042
\tf@size o193, o458, p282, p284
\TH l398, N179
\th l443, N179
\thanks *282*, F9
thebibliography (environment) *307*
\theequation z200, z212, z271, z307
\thefootnote G251, G287, G292, G312
\thempfn B216, G263, G268, G303, G308, G311
\thempfootnote B216, G253
\thepage I23, k73, w6, x14, x34, F143, H15, H32, J14, K192, K223, K1165
\Theta t218
\theta t194
\thicklines D46
\thickmuskip t532, z146
\thinlines D46
\thinmuskip i188, t530, z144, z147
\thinspace i188, i198, z119, z148
\thispagestyle J6
\thr@@ b16, p58, p208, p214, p227, p234, p241, p246, z223, z327, A222, A233, D165, D166, D168, D169, D201, D223
\tilde t427
\time a126, a130
\times t310
\title *282*, F3
\to t350
\today a131, a135, a143, a146, F8
\toks b31, b53, r381, r382, r392, r401, N229
\toks@ b41, i170, i171, i176, o123, o127, o129, o132, r6, r7, r192, r196, r202, r205, r210, r251, r252, r254, r255, r285, r287, r291,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

r308, r311, r370, r382, r383, r384, r430, r432, r438, r446, r450, r462, r465, r468, r476, r478, r803, r805, r807, r810, r812, r815, r818, r850, r851, L128, L129, L131, L132, L371, L372
\toksdef b46, b53
\tolerance b103, o480, J44, J52
\top t241
\topfigrule K545, K1547
\topfraction G171, K1513
\topmargin K52, K503
\topsep z312, A2, A59
\topskip b142, k49, A1, K109
\totalheight B17, B18, B19
\tr@ce . . . K188, K215, K274, K303, K348, K407, K422, K423, K424, K425, K436, K437, K438, K439, K440, K450, K599, K616, K632, K634, K720, K724, K736, K737, K738, K739, K745, K748, K756, K760, K771, K776, K781, K782, K783, K784, K791, K794, K802, K811, K817, K822, K827, K833, K834, K839, K844, K845, K846, K853, K857, K862, K866, K871, K882, K883, K885, K903, K912, K918, K927, K930, K936, K946, K950, K959, K965, K970, K975, K979, K983, K984, K991, K995, K999, K1006, K1015, K1019, K1023, K1024, K1028, K1029, K1035, K1041, K1051, K1057, K1061, K1062, K1067, K1068, K1074, K1077, K1078, K1079, K1086, K1087, K1088, K1096, K1100, K1112, K1113, K1120, K1123, K1131, K1135, K1139, K1140, K1144, K1145, K1201, K1206, K1217, K1227, K1234, K1244, K1289, K1302, K1303, K1307, K1310, K1312, K1315, K1318, K1320, K1361, K1368, K1373, K1379, K1384, K1388, K1394, K1402, K1403, K1410, K1415, K1420, K1422, K1428, K1430, K1437, K1464, K1465, K1474, K1490, K1495, K1503
\tracefloats K1244
\tracefloatvals K1244
\tracingall b248
\tracingcommands b249
\tracingfonts p17, p54, p58, p86, p116, p125, p148, p178, p192, p208, p214, p227, p234, p241, p246, p255, p268, p276, p279
\tracinglostchars b119, b250
\tracingmacros b251
\tracingoff p116, p276
\tracingon p117, p277
\tracingonline b240
\tracingoutput b243
\tracingpages b250
\tracingparagraphs b251
\tracingrestores b251
\tracingstats b249, N2
\triangle t247
\triangleleft t276, t390
\triangleright t277, t390
\trivlist y73, y80, y86, y100, z297, A89, C69, E35, E37
\try@load@fontshape o291, o331, p510, r141, r158
\try@simple@size . . . o312, p312, p453
\try@simples . . p401, p405, p412, p421
\try@size@range o320, p101, p312, p392
\try@size@substitution . . p103, p397
\try@sizes o20, o464
\tryif@simple p414, p420
\tryis@simple p419
\ttdefault s11, t29
\ttfamily s9, s10, v17, y120
\tw@ b16
\two@digits a33, a132, a133, d2, p498, L424, L455
\twocolumn K147
\type@restoreinfo p156, p161
\typein *24*, *24*, d18
\typeout *24*, a20, a63, a119, a144, a146, a158, a173, a180, a191, a204, a217, a230, a244, c13, c28, d3, d22, g77, k65, k172, k173, k178, k215, k225, k228, o285, s106, s113, s124, s135, s145, t9, t44, H8, H25, K1245, L95, N64, N183, N190, N202, N203, N211, O16

U

\u l192, l288, l368, l446, l453, l473, l480, l601
\uccode N30, N40, N49, N51, N55, N57, N135, N145, N154, N156, N160, N162

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

`\uchyph` b120
`\unboldmath` s51
`\UndeclareTextCommand` l146, l1095, l1096, l1097, l1098, l1099, l1100
`\undefinedpagestyle` J4, J8
`\underbar` b210
`\underbrace` t447
`\underline` b210, *233*, B259, B260
`\unhcopy` b212, C313, D391
`\unitlength` B33, B43, D5, D20, D21, D22, D23, D30, D31, D34, D35, D42, D78, D136, D189, D191, D204, D209, D211, D226, D228, D231, D268, D298, D309, D345, D346, D348, D349, D352, D353, D355, D356, D365, D367, D369, D371, N6
`\unlhd` s98
`\unpenalty` v99, y116
`\unrestored@protected@xdef` d227, G268, G292, G308, J21, J35
`\unrhd` s100
`\unvcopy` z123
`\Uparrow` t488
`\uparrow` t482
`\upbracefill` t450, t466
`\update@uclc@with@cyrillic` l814, l842, l866, l872
`\updefault` s20, t34, t41
`\Updownarrow` t492
`\updownarrow` t486
`\uplus` t292
`\upper@bound` . . p339, p340, p341, p355
`\uppercase` N169
`\upshape` l341, l579, l646, l728, s18, s19, s32, s77, v24
`\Upsilon` t223
`\upsilon` t205
`\use@mathgroup` o424, o442, o444, p253, r59, r352, r454, r457, r828, r852
`\usebox` B81
`\usecounter` A215, A228
`\usefont` o53, o230, s67, s80
`\usepackage` L202, L232
`\UseTextAccent` l115, l143, l963, l1091, l1093
`\UseTextSymbol` . l115, l141, l962, l1031

V

`\v` l193, l289, l367, l449, l450, l451, l455, l457, l460, l462, l464, l470, l476, l477, l478, l482, l484, l487, l489, l491, l497, l602
`\v@false` z78
`\v@true` z77, z79
`\vadjust` . i10, i38, i47, i139, i155, G126
`\valign` l955
`\value` I9, m14, *103*
`\varbigtriangledown` t280
`\varbigtriangleup` t281
`\varepsilon` j15, t210
`\varphi` t215
`\varpi` t212
`\varrho` t213
`\varsigma` t214
`\vartheta` t211
`\vbadness` b105
`\vdash` t318
`\vdots` t418
`\vec` t432
`\vector` g262, D133
`\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, r189, r190, r234, r254, r345, r383, r475, r808
`\version@list` r16, r21, r32, r182, r190, r239, r260, r279, r350, r395, r425, r480, r821
`\Vert` t477, t479
`\vert` t480
`\vfil` b199, l956, l958, D287, K144, K335, K506
`\vfilneg` b199
`\vfuzz` b128, J47, J54
`\vgl@` b189, b190
`\vglue` b189
`\vline` C334
`\voidb@x` b81, b220, n7
`\vphantom` z75, z94
`\vrule` . b193, i193, l247, l249, p144, t464, t465, t467, t468, B90, B92, B129, B136, B258, B288, C168, C201, C315, C334, D127, D177, D180, D196, D203, D218, D225, D287, D375, K1189, K1483

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltdefns.dtx`, e=`ltalloc.dtx`, f=`ltcntrl.dtx`, g=`lterror.dtx`, h=`ltpar.dtx`, i=`ltspace.dtx`, j=`ltlogos.dtx`, k=`ltfiles.dtx`, l=`ltoutenc.dtx`, m=`ltcounts.dtx`, n=`ltlength.dtx`, o=`ltfssbas.dtx`, p=`ltfsstrc.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=`ltfinal.dtx`, O=`ltpatch.ltx`

\vspace i132, i162, i163, i164
\vsplit K306

W

\warn@rel@i q15, q19, q71, q75, q80, q85, q109, q131
\wedge t282, t283
\widehat t435
\widetilde t434
\widowpenalty b112
\width B14
\wlog . . . a47, b40, b61, b69, L97, N236
\wp t232
\wr t296
\wrong@fontshape o295, o385

X

\x o250, o251
\x@protect d205, d216
\Xi t220
\xi t200

Y

\year a132, c6, L455
\yxdim D323

Z

\Z N44, N122, N149
\z N33, N123, N138
\z@ b81
\z@skip b81
\zap@space k84, L117, L265, L282, L299
\zeta t192