The LaTeX 2_ε Sources

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

2020-10-01 Patch level 2

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

Contents

a ltdirchk.dtx **1**

1 LaTeX System Dependent Initialisations **1**

2 Initialisation **2**
2.1 INITEX 2
2.2 Some bits of 2e 4

3 texsys.cfg **5**
3.1 texsys.cfg 5
3.2 UNIX (web2c) 6
3.3 UNIX (other) 7
3.4 MSDOS (emtex) 7
3.5 MSDOS (other) 7
3.6 VMS (DECUS TeX, PD VMS 3.6) 7
3.7 VMS (???) 7
3.8 MACINTOSH (OzTeX 1.6) 8
3.9 MACINTOSH (other) 8
3.10 FAKE EXAMPLE 8

4 Setting `\@currdir` **9**

5 Setting `\input@path` **10**

6 Filename Parsing 11

7 TEX Versions 13

8 ltxcheck.tex 13

b ltplain.dtx 14

1 Plain TEX 14

c ltvers.dtx 34

1 Version Identification 34

d ltluatex.dtx 38

1 Overview 38

2 Core TEX functionality 38

3 Plain TEX interface 39

4 Lua functionality 39
4.1 Allocators in Lua . . . 39
4.2 Lua access to TEX register numbers . . . 40
4.3 Module utilities . . . 41
4.4 Callback management . . . 41

5 Implementation 42
5.1 Minimum LuaTEX version . . . 42
5.2 Older LATEX/Plain TEX setup . . . 42
5.2.1 Fixes to `etex.src/etex.sty` . . . 43
5.2.2 luatex specific settings . . . 43
5.3 Attributes . . . 44
5.4 Category code tables . . . 44
5.5 Named Lua functions . . . 46
5.6 Custom whatsits . . . 47
5.7 Lua bytecode registers . . . 47
5.8 Lua chunk registers . . . 47
5.9 Lua loader . . . 47
5.10 Lua module preliminaries . . . 49
5.11 Lua module utilities . . . 49
5.11.1 Module tracking . . . 49
5.11.2 Module messages . . . 50
5.12 Accessing register numbers from Lua . . . 51
5.13 Attribute allocation . . . 52
5.14 Custom whatsit allocation . . . 53
5.15 Bytecode register allocation . . . 53

5.16 Lua chunk name allocation . . . 54
5.17 Lua function allocation . . . 54
5.18 Lua callback management . . . 54
5.18.1 Housekeeping . . . 54
5.18.2 Handlers . . . 57
5.18.3 Public functions for callback management . . . 59

e ltexpl.dtx 65

1 expl3-dependent code 65
1.1 Loader . . . 65
1.2 Using expl3 code . . . 68

f ltdefns.dtx 70

1 Definitions 70
1.1 Initex initialisations . . . 70
1.2 Saved versions of TEX primitives . . . 70
1.3 Command definitions . . . 71
1.4 Robust commands and protect . . . 80
1.5 Acting on robust commands . . . 86
1.5.1 Copying robust commands . . . 88
1.5.2 Showing robust commands . . . 89
1.5.3 Commands defined with `\DeclareRobustCommand` . . . 90
1.5.4 Commands defined with `\newcommand` (with optional argument) . . . 92
1.6 Internal defining commands . . . 93

2 Discretionary Hyphenation 97

g lthooks.dtx 100

1 Introduction 100

2 Package writer interface 100
2.1 LATEX 2_ε interfaces . . . 100
2.1.1 Declaring hooks and using them in code . . . 100
2.1.2 Updating code for hooks . . . 102
2.1.3 Hook names and default labels . . . 103
2.1.4 Defining relations between hook code . . . 104
2.1.5 Querying hooks . . . 106
2.1.6 Displaying hook code . . . 107
2.1.7 Debugging hook code . . . 109
2.2 L3 programming layer (`expl3`) interfaces . . . 109
2.3 On the order of hook code execution . . . 111
2.4 The use of “reversed” hooks . . . 112
2.5 Difference between “normal” and “one-time” hooks . . . 113
2.6 Private LATEX kernel hooks . . . 114
2.7 Legacy LATEX 2_ε interfaces . . . 114

2.8 LaTeX 2ε commands and environments augmented by hooks 115
2.8.1 Generic hooks for all environments 115
2.8.2 Hooks provided by `\begin{document}` 116
2.8.3 Hooks provided by `\end{document}` 117
2.8.4 Hooks provided `\shipout` operations 118
2.8.5 Hooks provided by file loading operations 119
2.8.6 Hooks provided in NFSS commands 119

3 The Implementation 119
3.1 Loading further extensions 119
3.2 Debugging 120
3.3 Borrowing from internals of other kernel modules 120
3.4 Declarations 120
3.5 Providing new hooks 122
3.6 Parsing a label 124
3.7 Setting rules for hooks code 132
3.8 Specifying code for next invocation 145
3.9 Using the hook 146
3.10 Querying a hook 148
3.11 Messages 150
3.12 LaTeX 2ε package interface commands 150
3.13 Internal commands needed elsewhere 153

h ltalloc.dtx 155

1 Counters 155

i ltcntrl.dtx 157

1 Program control structure 157

j lterror.dtx 161

1 Error handling and tracing 161
1.1 General commands 161
1.2 Specific errors 166
1.3 Tracing 170

k ltpar.dtx 171

1 Paragraphs 171
1.1 Implementation 171

l ltspace.dtx 173

1 Spacing **173**
1.1 User Commands . . . 173
1.2 Chris' comments . . . 173
1.3 Some immediate actions . . . 175
1.4 The code . . . 176
1.5 Vertical spacing . . . 183
1.6 Horizontal space (and breaks) . . . 188

m ltlogos.dtx **192**

1 Logos **192**

n ltfiles.dtx **193**

1 File Handling **193**
1.1 Safe Input Macros . . . 205
1.2 Listing files . . . 211

o ltoutenc.dtx **213**

1 Font encodings **213**
1.1 Removing encoding-specific commands . . . 215
1.2 The order of declarations . . . 216
1.3 Docstrip modules . . . 216
1.4 Definitions for the kernel . . . 217
1.4.1 Declaration commands . . . 217
1.4.2 Hyphenation . . . 225
1.4.3 Miscellania . . . 225
1.4.4 Default encodings . . . 225
1.4.5 Math material . . . 228
1.5 Definitions for the OT1 encoding . . . 229
1.6 Definitions for the T1 encoding . . . 231
1.7 Definitions for the OMS encoding . . . 237
1.8 Definitions for the OML encoding . . . 237
1.9 Definitions for the OT4 encoding . . . 238
1.10 Definitions for the TS1 encoding . . . 240
1.11 Definitions for the TU encoding . . . 244

2 Package files **255**
2.1 The fontenc package . . . 255

p ltcounts.dtx **258**

1 Counters and Lengths **258**
1.1 Environment Counter Macros . . . 258

q ltlength.dtx 266

1 Lengths 266

r ltfssbas.dtx 268

1 Preliminary macros 268

2 Macros for setting up the tables 269

3 Selecting a new font 276
3.1 Macros for the user . . . 276
3.2 Macros for loading fonts . . . 280

4 Assigning math fonts to *versions* 287

s ltfssaxes.dtx 294

1 Changing the font series 294
1.1 The series lookup table . . . 294
1.2 Mapping rules for series changes . . . 295
1.3 Changing to a new series . . . 303

2 Changing the shape 306
2.1 Mapping rules for shape combinations . . . 307
2.2 Changing to a new shape . . . 308

3 Make sure we win . . . 309

t ltfsstrc.dtx 312

1 Introduction 312

2 A driver for this document 312

3 The Implementation 313

4 Handling Options 313

5 Macros common to `fam.tex` and `tracefnt.sty` 315
5.1 General font loading . . . 315
5.2 Math fonts setup . . . 320
5.2.1 Outline of algorithm for math font sizes . . . 320
5.2.2 Code for math font size setting . . . 321
5.2.3 Other code for math . . . 322

6 Scaled font extraction 324
6.1 Sizefunctions . . . 331

u ltfsscmp.dtx **335**

v ltfssdcl.dtx **340**

1 Interface Commands **340**

w ltfssini.dtx **367**

1 NFSS Initialisation **367**
1.1 Providing math *versions* 367

2 Custom series settings for main document families **368**

3 Supporting nested emphasis **382**
3.1 Legacy 385
3.2 Miscellaneous 386

x fontdef.dtx **390**

1 Introduction **390**

2 Customization **390**

3 The `docstrip` modules **391**

4 A driver for this document **391**

5 The `fonttext.ltx` file **391**
5.1 Encodings 392
5.2 Defaults 394

6 The `fontmath.ltx` file **396**
6.1 The font encodings used 396
6.1.1 Symbolfont and Alphabet declarations 396
6.2 Math font sizes 397
6.3 The math symbol assignments 398
6.3.1 The letters 398
6.3.2 The digits 399
6.3.3 Punctuation, brace, etc. keys 399
6.3.4 Delimitercodes for characters 399
6.4 Symbols accessed via control sequences 400
6.4.1 Greek letters 400
6.4.2 Ordinary symbols 401
6.4.3 Large Operators 401
6.4.4 Binary symbols 402
6.4.5 Relations 403
6.4.6 Arrows 404
6.4.7 Punctuation symbols 405
6.4.8 Math accents 406

6.4.9 Radicals . . . 406
6.4.10 Over and under something, etc . . . 406
6.4.11 Delimiters . . . 407
6.5 Math versions of text commands . . . 408
6.6 Other special functions and parameters . . . 408
6.6.1 Biggggg . . . 408
6.6.2 The log-like functions . . . 409
6.6.3 Parameters . . . 409

7 Default cfg files 409

y preload.dtx 411

1 Overview 411

2 Customization 411

3 Module switches for the DOCSTRIP program 411

4 A driver for this document 412

5 The code 412

z ltfntcmd.dtx 414

1 Introduction 414

2 The implementation 416

3 Initialization 422

A lttextcomp.dtx 423

1 Sub-encodings 427
1.1 Sub-encoding 1 (drop symbols not working in Latin Modern) . . . 429
1.2 Sub-encoding 2 (majority of new OTF fonts via autoinst) . . . 429
1.3 Sub-encoding 3 . . . 431
1.4 Sub-encoding 4 . . . 431
1.5 Sub-encoding 5 (most older PS fonts) . . . 431
1.6 Sub-encoding 6 . . . 432
1.7 Sub-encoding 7 . . . 432
1.8 Sub-encoding 8 . . . 432
1.9 Sub-encoding 9 (most missing) . . . 432

2 Unicode engine specials 432

3 Font family sub-encodings setup 433

4 Legacy symbol support for lists and footnote symbols 437

5 The textcomp package **441**
5.1 The old textcomp package code . . . 442
5.1.1 Supporting oldstyle digits . . . 451
5.1.2 Subset encoding defaults . . . 451

B ltpageno.dtx **454**

1 Page Numbering **454**

C ltxref.dtx **455**

1 Cross Referencing **455**
1.1 Cross Referencing . . . 455

D ltmiscen.dtx **460**

1 Miscellaneous Environments **460**
1.1 Environments . . . 460
1.2 Center, Flushright, Flushleft . . . 471
1.3 Verbatim . . . 474

E ltmath.dtx **481**

1 Math setup **481**
1.1 Math commands based on plain TEX . . . 481
1.1.1 The log-like functions . . . 481
1.1.2 Biggggg . . . 482
1.1.3 The UNSORTED Rest . . . 482
1.2 Math Environments . . . 488
1.3 External options to the standard document classes . . . 493
1.3.1 Left equation numbering . . . 493
1.3.2 Flush left equations . . . 493

F ltlists.dtx **496**

1 List, and related environments **496**
1.1 List and Trivlist . . . 497
1.2 Vertical Spacing (skips) . . . 498
1.3 Penalties . . . 498
1.4 Horizontal Spacing (dimens) . . . 498
1.5 Default Values . . . 498
1.6 Itemize and Enumerate . . . 509

G ltboxes.dtx **512**

1 LaTeX Box commands 512
1.1 Some low-level constructs 526

H lttab.dtx 527

1 Tabbing, Tabular and Array Environments 527
1.1 tabbing 527
1.2 array and tabular environments 536

I ltpictur.dtx 552

1 Picture Mode 552
1.1 Curves 579

J ltthm.dtx 584

1 Theorem Environments 584

K ltsect.dtx 588

1 Sectioning Commands 588
1.1 The Title 588
1.2 Sectioning 589
1.2.1 Initializations 596
1.3 Table of Contents etc. 596
1.3.1 Convention 596
1.3.2 Commands 596

L ltfloat.dtx 601

1 Floats 601
1.1 Floating Environments 601
1.2 Footnotes 615

M ltidxglo.dtx 622

1 Index and Glossary Generation 622

N ltbibl.dtx 625

1 Bibliography Generation 625
1.1 Default definitions 628

O ltpage.dtx 629

1 Page styles and related commands **629**
1.1 Page Style Commands . . . 629
1.2 How a page style makes running heads and feet . . . 629
1.3 marking conventions . . . 629

P ltclass.dtx **633**

1 Introduction **633**

2 User interface **633**
2.1 Option processing . . . 634

3 Class and Package interface **635**
3.1 Class name and version . . . 635
3.2 Package name and version . . . 635
3.3 Requiring other packages . . . 635
3.4 Declaring new options . . . 636
3.5 Safe Input Macros . . . 637

4 Implementation **637**
4.1 Hooks . . . 660
4.2 Providing shipment . . . 662

5 Package/class rollback mechanism **670**

6 After Preamble **678**

Q ltfilehook.dtx **679**

1 Introduction **679**
1.1 Provided hooks . . . 679
1.2 General hooks for file reading . . . 679
1.3 Hooks for package and class files . . . 680
1.4 Hooks for `\include` files . . . 681
1.5 High-level interfaces for LaTeX . . . 682
1.6 Internal interfaces for LaTeX . . . 682
1.7 A sample package for structuring the log output . . . 683

2 The Implementation **684**
2.1 Document and package-level commands . . . 684
2.2 expl3 helpers . . . 684
2.3 Declaring the file-related hooks . . . 687
2.4 Patching LaTeX's `\InputIfFileExists` command . . . 687
2.5 Declaring a file substitution . . . 689
2.6 Selecting a file (`\set@curr@file`) . . . 691
2.7 Replacing a file and detecting loops . . . 694
2.7.1 The Tortoise and Hare algorithm . . . 694
2.8 Preventing a package from loading . . . 696
2.9 High-level interfaces for LaTeX . . . 697

2.10 Internal commands needed elsewhere 697

3 A sample package for structuring the log output 698

4 Package emulations 699
4.1 Package atveryend emulation 699

R ltshipout.dtx 700

1 Introduction 700
1.1 Overloading the \shipout primitive 700
1.2 Provided hooks 701
1.3 Special commands for use inside the hooks 702
1.4 Information counters 703
1.5 Debugging shipout code 703

2 Emulating commands from other packages 703
2.1 Emulating atbegshi 704
2.2 Emulating everyshi 704
2.3 Emulating atenddvi 705
2.4 Emulating everypage 705

3 The Implementation 705
3.1 Debugging 705
3.2 Handling the end of job hook 714

4 Legacy LaTeX 2ε interfaces 717

5 Internal commands needed elsewhere 717

6 Package emulation for compatibility 719
6.1 Package atenddvi emulation 719
6.2 Package atbegshi emulation 719
6.3 Package everyshi emulation 721

S ltoutput.dtx 722

1 Output Routine 722
1.1 Floats 722
1.1.1 Kludgeins 778
1.1.2 Float control 779
1.1.3 Float placement parameters 792

T lthyphen.dtx 795

U ltfinal.dtx 797

1 Final settings **797**
1.1 Debugging . . . 797
1.2 Typesetting parameters . . . 797
1.3 Lccodes for hyphenation . . . 799
1.4 Hyphenation . . . 802
1.5 Font loading . . . 803
1.6 Input encoding . . . 803
1.7 Lccodes and uccodes . . . 807
1.8 Applying Patch files . . . 809
1.9 Freeing Memory . . . 810
1.10 Initialise file list . . . 811
1.11 Do some temporary work for pre-release . . . 811
1.12 Some last minute initializations 811
1.13 Dumping the format . . . 811

Change History **812**

Index **875**

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`,`\filenam` 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.
...

.3 \renewcommand{\rubbish}
                          {}
```

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

```
 LaTeX Error: \rubbish undefined.

ee the LaTeX manual or LaTeX Companion for explanation.
ype  H <return>  for immediate help.
 .
...

.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
```

[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.

```
\catcode`\}=2
```

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

```
\ifx\directlua\undefined
\else
  \ifx\luatexversion\undefined
```

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

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

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

```
⟨/initex⟩
⟨/dircheck⟩
⟨*initex, latexrelease⟩
⟨latexrelease⟩\ifx\directlua\undefined\else
⟨latexrelease⟩\IncludeInRelease{2015/10/01}{\luatexluafunction}
⟨latexrelease⟩                              {LuaTeX (prefixed names)}%
    \directlua{tex.enableprimitives("",%
                  tex.extraprimitives("omega", "aleph", "luatex"))}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\luatexluafunction}
⟨latexrelease⟩                              {LuaTeX (prefixed names)}%
⟨latexrelease⟩\directlua{
⟨latexrelease⟩  tex.enableprimitives(
⟨latexrelease⟩    "luatex",
⟨latexrelease⟩    tex.extraprimitives("core","omega", "aleph", "luatex")
⟨latexrelease⟩  )
⟨latexrelease⟩  local i
⟨latexrelease⟩  local t = { }
⟨latexrelease⟩  for _,i in pairs(tex.extraprimitives("luatex")) do
⟨latexrelease⟩    if not string.match(i,"^U") then
⟨latexrelease⟩      if not string.match(i, "^luatex") then
⟨latexrelease⟩        table.insert(t,i)
⟨latexrelease⟩      end
⟨latexrelease⟩    else
⟨latexrelease⟩      if string.match(i,"^Uchar$") then
⟨latexrelease⟩        table.insert(t,i)
⟨latexrelease⟩      end
⟨latexrelease⟩    end
⟨latexrelease⟩  end
⟨latexrelease⟩  for _,i in pairs(t) do
⟨latexrelease⟩    tex.print(
⟨latexrelease⟩      "\noexpand\\let\noexpand\\" .. i
⟨latexrelease⟩        .. "\noexpand\\undefined"
⟨latexrelease⟩    )
⟨latexrelease⟩  end
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\fi
⟨/initex, latexrelease⟩
⟨*dircheck⟩
⟨*initex⟩
```

```
  \fi
\fi
```

A test can now be made for eTEX.

```
⟨initex⟩\ifx\eTeXversion\undefined
⟨initex⟩   \errmessage
⟨initex⟩     {LaTeX requires e-TeX}
⟨initex⟩   \expandafter\endinput
⟨initex⟩\fi
```

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

```
\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}
\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⟩
```

(*End definition for* `\strip@prefix`*. This function is documented on page* **??**.)

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
\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,\filenam 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 compatibility with macro sets, you want `name.ext` to be mapped to `<ext>name`. and `<area>name.ext` to be mapped to `<area.ext>name`. `\input` does this mapping automatically, but `\openin` does not, and does not look in the same places as `\input`. `<>name` is the desired 'current directory' syntax.

the following code would possibly work:

```
% \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
```

The current date and time stamp.

\today

```
\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. Note that this definition is overwritten later on again!

```
\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%
         }
```

(*End definition for* `\@currdir`*,* `\IfFileExists`*, and* `\today`*. These functions are documented on page* ??.)

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 2_ε can not be expected to have enough information to supply the correct information to the prompts. Now the interaction is omitted. After the format is made the installer can attempt to run the test document `ltxcheck.tex` through LaTeX 2_ε. This will check, amongst other things, whether `texsys.cfg` will need to be edited and the format remade.

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

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

```
  \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
    }
```

(*End definition for* `\input@path`*. This function is documented on page* ??.)

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.

```
⟨/dircheck⟩
⟨*dircheck, latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}{\filename@simple}
⟨latexrelease⟩                                {Final dot for extension}%
\def\filename@simple#1.#2\\{%
  \ifx\\#2\\%
    \let\filename@ext\relax
    \edef\filename@base{#1}%
  \else
    \filename@dots{#1}#2\\%
  \fi}
\def\filename@dots#1#2.#3\\{%
  \ifx\\#3\\%
    \def\filename@ext{#2}%
    \edef\filename@base{#1}%
  \else
    \filename@dots{#1.#2}#3\\%
  \fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\filename@simple}
⟨latexrelease⟩                                {Final dot for extension}%
⟨latexrelease⟩   \def\filename@simple#1.#2\\{%
⟨latexrelease⟩     \ifx\\#2\\%
⟨latexrelease⟩        \let\filename@ext\relax
```

```
⟨latexrelease⟩      \else
⟨latexrelease⟩         \edef\filename@ext{\filename@dot#2\\}%
⟨latexrelease⟩      \fi
⟨latexrelease⟩      \edef\filename@base{#1}}
⟨latexrelease⟩ \EndIncludeInRelease
⟨/dircheck, latexrelease⟩
⟨*dircheck⟩
```

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
```

(*End definition for* `\filename@parse`*. This function is documented on page* **??**.)

7 TEX Versions

`\@TeXversion` TEX versions older than 3.141 require `\@TeXversion` to be set. This can be determined automatically due to a trick suggested by Bernd Raichle. (Actually this will not always get the correct version number, eg 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
```

(*End definition for* `\@TeXversion`*. This function is documented on page* **??**.)

```
⟨/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

1 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⟩
\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 \def^^L{\par}% ascii form-feed is \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
```

(*End definition for* `\@ne` *and others. These functions are documented on page* **??**.)

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

Constants above 255 defined using `\mathchardef`.

```
\mathchardef\@cclvi=256
\mathchardef\@m=1000
\mathchardef\@M=10000
\mathchardef\@MM=20000
```

(*End definition for* `\@cclvi` *and others. These functions are documented on page* **??**.)

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.

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

The following counters are reserved:

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

22 constant -1

End of historical LaTeX 2.09 comments.

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

(*End definition for* `\insc@unt` *and* `\allocationnumber`*. These functions are documented on page* **??**.)

`\m@ne`

The constant -1.

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

(*End definition for* `\m@ne`*. This function is documented on page* **??**.)

`\wlog`

Write on log file (only)

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

(*End definition for* `\wlog`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\count@` *and others. These functions are documented on page* ??.)

`\newcount` `\newdimen` `\newskip` `\newmuskip` `\newbox` `\newtoks` `\newread` `\newwrite` `\newfam` `\newlanguage`

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.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\newcount}{Extended Allocation}%
```

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

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

```
\def\newbox   {\e@alloc\box
                  {\ifnum\allocationnumber<\@cclvi
                     \expandafter\chardef
                   \else
                     \expandafter\e@alloc@chardef
                   \fi}
                                          {\count14}\insc@unt\float@count}
\def\newtoks  {\e@alloc\toks \toksdef{\count15}\m@ne\e@alloc@top}
\def\newread  {\e@alloc\read \chardef{\count16}\m@ne\sixt@@n}
```

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

```
\ifx\directlua\@undefined
  \def\newwrite   {\e@alloc\write \chardef{\count17}\m@ne\sixt@@n}
\else
  \def\newwrite   {\e@alloc\write
                   {\ifnum\allocationnumber=18
                     \advance\count17\@ne
                     \allocationnumber\count17 %
                    \fi
                    \global\chardef}%
                   {\count17}%
                   \m@ne
                   {128}}
\fi
```

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

```
\ifx\directlua\@undefined
  \def\newlanguage  {\e@alloc\language \chardef{\count19}\m@ne\@cclvi}
\else
  \def\newlanguage  {\e@alloc\language \chardef{\count19}\m@ne{16384}}
```

```
\fi
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\newcount}{Extended Allocation}%
⟨latexrelease⟩\def\newcount{\alloc@0\count\countdef\insc@unt}
⟨latexrelease⟩\def\newdimen{\alloc@1\dimen\dimendef\insc@unt}
⟨latexrelease⟩\def\newskip{\alloc@2\skip\skipdef\insc@unt}
⟨latexrelease⟩\def\newmuskip{\alloc@3\muskip\muskipdef\@cclvi}
⟨latexrelease⟩\def\newbox{\alloc@4\box\chardef\insc@unt}
⟨latexrelease⟩\def\newtoks{\alloc@5\toks\toksdef\@cclvi}
⟨latexrelease⟩\def\newread{\alloc@6\read\chardef\sixt@@n}
⟨latexrelease⟩\def\newwrite{\alloc@7\write\chardef\sixt@@n}
⟨latexrelease⟩\def\new@mathgroup{\alloc@8\fam\chardef\sixt@@n}
⟨latexrelease⟩\def\newlanguage{\alloc@9\language\chardef\@cclvi}
⟨latexrelease⟩\let\newfam\new@mathgroup
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\newcount` *and others. These functions are documented on page* **??**.)

`\e@alloc@chardef`
`\e@alloc@top`

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

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\e@alloc@chardef}{Extended Allocation}%
```

```
\ifx\directlua\@undefined
  \ifx\widowpenalties\@undefined
```

classic tex has 2^8 registers.

```
    \mathchardef\e@alloc@top=255
    \let\e@alloc@chardef\chardef
  \else
```

etex and xetex have 2^{15} registers.

```
    \mathchardef\e@alloc@top=32767
    \let\e@alloc@chardef\mathchardef
  \fi
\else
```

luatex has 2^{16} registers.

```
  \chardef\e@alloc@top=65535
  \let\e@alloc@chardef\chardef
\fi
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\e@alloc@chardef}{Extended Allocation}%
⟨latexrelease⟩\let\e@alloc@top\@undefined
⟨latexrelease⟩\let\e@alloc@chardef\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\e@alloc@chardef` *and* `\e@alloc@top`*. These functions are documented on page* **??**.)

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

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\e@mathgroup@top}{Extended Allocation}%
```

```
\ifx\Umathcode\@undefined
```

classic and e tex have 16 fam (0–15).

```
  \chardef\e@mathgroup@top=16
\else
```

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

```
  \chardef\e@mathgroup@top=256
\fi
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\e@mathgroup@top}{Extended Allocation}%
⟨latexrelease⟩\let\e@mathgroup@top\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\e@mathgroup@top`. *This function is documented on page* **??**.)

`\e@alloc` A modified version of `\alloc@` that takes the count register rather than just the final digit of its number (assuming `\count1`x). It also has an extra argument to give the top of the extended range.

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

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

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\e@alloc}{Extended Allocation}%
```

```
\def\e@alloc#1#2#3#4#5#6{%
  \global\advance#3\@ne
  \e@ch@ck{#3}{#4}{#5}#1%
  \allocationnumber#3\relax
  \global#2#6\allocationnumber
  \wlog{\string#6=\string#1\the\allocationnumber}}%
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\e@alloc}{Extended Allocation}%
⟨latexrelease⟩\let\e@alloc\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\e@alloc`. *This function is documented on page* **??**.)

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

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

`\extrafloats`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/10/01}
⟨latexrelease⟩                  {\e@ch@ck}{Extended Allocation (checking)}%
\gdef\e@ch@ck#1#2#3#4{%
  \ifnum#1<#2\else
```

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

```
    \ifnum#1=#2\relax
      \global#1\@cclvi
      \ifx\count#4\global\advance#1 10 \fi
    \fi
```

Check we are below the extended limit.

```
    \ifnum#1<#3\relax
    \else
      \errmessage{No room for a new \string#4}%
    \fi
  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\e@ch@ck}{Extended Allocation (checking)}%
⟨latexrelease⟩\gdef\e@ch@ck#1#2#3#4{%
⟨latexrelease⟩  \ifnum#1<#2\else
⟨latexrelease⟩    \ifnum#1=#2\relax
⟨latexrelease⟩      #1\@cclvi
⟨latexrelease⟩      \ifx\count#4\advance#1 10 \fi
⟨latexrelease⟩    \fi
⟨latexrelease⟩    \ifnum#1<#3\relax
⟨latexrelease⟩    \else
⟨latexrelease⟩      \errmessage{No room for a new #4}%
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\e@ch@ck}{Extended Allocation (checking)}%
⟨latexrelease⟩\let\e@ch@ck\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\extrafloats}{Extra floats}%
\let\float@count\e@alloc@top
```

`\extrafloats`

```
\ifx\numexpr\@undefined
```

In classic TeX use `\newinsert` to allocate float boxes.

```
\def\extrafloats#1{%
\count@#1\relax
\ifnum\count@>\z@
\newinsert\reserved@a
\global\expandafter\chardef
```

```
            \csname bx@\the\allocationnumber\endcsname\allocationnumber
\@cons\@freelist{\csname bx@\the\allocationnumber\endcsname}%
\advance\count@\m@ne
\expandafter\extrafloats
\expandafter\count@
\fi
}%
```

```
\else
```

In e-tex take float boxes from the top of the extended range.

```
\def\extrafloats#1{%
\ifnum#1>\z@
\count@\numexpr\float@count-1\relax
  \ch@ck0\count@\count
  \ch@ck1\count@\dimen
  \ch@ck2\count@\skip
  \ch@ck4\count@\box
\global\e@alloc@chardef\float@count\count@
\global\expandafter\e@alloc@chardef
            \csname bx@\the\float@count\endcsname\float@count
\@cons\@freelist{\csname bx@\the\float@count\endcsname}%
\expandafter
\extrafloats\expandafter{\numexpr#1-1\relax}%
\fi}%
```

```
\fi
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\extrafloats}{Extra floats}%
⟨latexrelease⟩\let\float@count\@undefined
⟨latexrelease⟩\let\extrafloats\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\e@ch@ck`*,* `\extrafloats`*, and* `\extrafloats`*. These functions are documented on page* ??.)

`\alloc@` Since `\e@alloc` was added in 2015, `\@alloc` has not been used, but was left as some legacy code calls it. However the original defnition gives spurious errors once the "classic" registers run out, so it is now defined to call `\e@alloc` internally.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}
⟨latexrelease⟩                  {\alloc@}{emulate alloc@}%
\def\alloc@#1#2#3#4{\e@alloc#2#3{\count1#1}#4\float@count}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\alloc@}{emulate alloc@}%
⟨latexrelease⟩\def\alloc@#1#2#3#4#5{\global\advance\count1#1\@ne
⟨latexrelease⟩  \ch@ck#1#4#2%
⟨latexrelease⟩  \allocationnumber\count1#1%
⟨latexrelease⟩  \global#3#5\allocationnumber
```

```
⟨latexrelease⟩   \wlog{\string#5=\string#2\the\allocationnumber}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\alloc@`*. This function is documented on page* **??**.)

`\newinsert`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/10/01}
⟨latexrelease⟩                    {\newinsert}{Extended \newinsert}%
\ifx\numexpr\@undefined
```

If e-TEX is not available use the original plain TEX definition of `\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}}
\else
```

The highest register allowed with `\insert`.

```
\ifx\directlua\@undefined
  \chardef\e@insert@top255
\else
  \chardef\e@insert@top\e@alloc@top
\fi
```

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

```
\def\newinsert#1{%
\@tempswafalse
\global\advance\insc@unt\m@ne
\ifnum\count10<\insc@unt
\ifnum\count11<\insc@unt
\ifnum\count12<\insc@unt
\ifnum\count14<\insc@unt
  \@tempswatrue
\fi\fi\fi\fi
\if@tempswa
\allocationnumber\insc@unt
\else
\global\advance\insc@unt\@ne
  \extrafloats\@ne
  \@next\@currbox\@freelist
    {\ifnum\@currbox<\e@insert@top
      \allocationnumber\@currbox
     \else
     \ch@ck0\m@ne\insert
     \fi}%
     {\ch@ck0\m@ne\insert}%
```

```
\fi
\global\chardef#1\allocationnumber
\wlog{\string#1=\string\insert\the\allocationnumber}%
}
```

```
\fi
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\newinsert}{Extended \newinsert}%
⟨latexrelease⟩\let\e@insert@top\@undefined
⟨latexrelease⟩\def\newinsert#1{\global\advance\insc@unt \m@ne
⟨latexrelease⟩  \ch@ck0\insc@unt\count
⟨latexrelease⟩  \ch@ck1\insc@unt\dimen
⟨latexrelease⟩  \ch@ck2\insc@unt\skip
⟨latexrelease⟩  \ch@ck4\insc@unt\box
⟨latexrelease⟩  \allocationnumber\insc@unt
⟨latexrelease⟩  \global\chardef#1\allocationnumber
⟨latexrelease⟩  \wlog{\string#1=\string\insert\the\allocationnumber}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\newinsert`*. This function is documented on page* **??**.)

`\ch@ck`

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

(*End definition for* `\ch@ck`*. This function is documented on page* **??**.)

`\newhelp`

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

(*End definition for* `\newhelp`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\maxdimen` *and* `\hideskip`*. These functions are documented on page* **??**.)

`\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
```

(*End definition for* `\p@` *and others. These functions are documented on page* **??**.)

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. *Historical LATEX 2.09 comments (not necessarily accurate any more):*

```
\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
```

End of historical LaTeX 2.09 comments.

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
```

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

```
\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
```

End of historical LaTeX 2.09 comments.

\normalbaselineskip
\normallineskip
\normallineskiplimit

We also define special registers that function like parameters:

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

(*End definition for* `\normalbaselineskip`, `\normallineskip`, *and* `\normallineskiplimit`. *These functions are documented on page* **??**.)

\interfootlinepenalty

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

(*End definition for* `\interfootlinepenalty`. *This function is documented on page* **??**.)

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}
```

(*End definition for* `\magstephalf` *and* `\magstep`. *These functions are documented on page* **??**.)

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 }
```

(*End definition for* `\frenchspacing` *and* `\nonfrenchspacing`. *These functions are documented on page* **??**.)

\normalbaselines

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

(*End definition for* `\normalbaselines`. *This function is documented on page* **??**.)

\M
\I

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

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

(*End definition for* `\M` *and* `\I`. *These functions are documented on page* **??**.)

\lq
\rq

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

(*End definition for* `\lq` *and* `\rq`. *These functions are documented on page* **??**.)

\lbrack
\rbrack

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

(*End definition for* `\lbrack` *and* `\rbrack`*. These functions are documented on page* **??**.)

\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}
```

(*End definition for* `\aa` *and* `\AA`*. These functions are documented on page* **??**.)

\endgraf
\endline

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

(*End definition for* `\endgraf` *and* `\endline`*. These functions are documented on page* **??**.)

\space

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

(*End definition for* `\space`*. This function is documented on page* **??**.)

\empty

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

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

(*End definition for* `\empty`*. This function is documented on page* **??**.)

\null

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

(*End definition for* `\null`*. This function is documented on page* **??**.)

\bgroup
\egroup

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

(*End definition for* `\bgroup` *and* `\egroup`*. These functions are documented on page* **??**.)

\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}
```

(*End definition for* `\obeylines` *and* `\obeyspaces`*. These functions are documented on page* **??**.)

\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
```

(*End definition for* `\loop`, `\iterate`, *and* `\repeat`. *These functions are documented on page* ??.)

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

\nointerlineskip
\offinterlineskip

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

(*End definition for* `\nointerlineskip` *and* `\offinterlineskip`. *These functions are documented on page* ??.)

\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@}
```

(*End definition for* `\vglue` *and* `\hglue`. *These functions are documented on page* ??.)

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

\slash

This generates a `/` acting a bit like `-` but still allows hyphenation in the word part preceding it (but not after).

```
\def\slash{/\penalty\exhyphenpenalty}
```

(*End definition for* `\slash`. *This function is documented on page* ??.)

\break
\nobreak
\allowbreak

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

(*End definition for* `\break`, `\nobreak`, *and* `\allowbreak`. *These functions are documented on page* ??.)

\filbreak
\goodbreak

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

(*End definition for* `\filbreak` *and* `\goodbreak`. *These functions are documented on page* ??.)

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

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

(*End definition for* `\eject`*. This function is documented on page* **??**.)

`\removelastskip`

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

(*End definition for* `\removelastskip`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\smallbreak`*,* `\medbreak`*, and* `\bigbreak`*. These functions are documented on page* **??**.)

`\m@th`

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

(*End definition for* `\m@th`*. This function is documented on page* **??**.)

`\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@}}
```

(*End definition for* `\underbar`*. This function is documented on page* **??**.)

`\strutbox` `\strut` LATEX sets `\strutbox` in `\set@fontsize`.

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

(*End definition for* `\strutbox` *and* `\strut`*. These functions are documented on page* **??**.)

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

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

(*End definition for* `\hidewidth`*. This function is documented on page* **??**.)

`\narrower`

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

(*End definition for* `\narrower`*. This function is documented on page* **??**.)

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}
```

(*End definition for* `\leavevmode`*. This function is documented on page* ??.)

\mathhexbox

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

(*End definition for* `\mathhexbox`*. This function is documented on page* ??.)

\ialign

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

(*End definition for* `\ialign`*. This function is documented on page* ??.)

\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}
```

(*End definition for* `\oalign`*,* `\o@lign`*, and* `\ooalign`*. These functions are documented on page* ??.)

\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
```

(*End definition for* `\sh@ft`*. This function is documented on page* ??.)

\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
```

(*End definition for* `\ltx@sh@ft`*. This function is documented on page* ??.)

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 beginning 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@}
```

(*End definition for* \hrulefill *and* \dotfill*. These functions are documented on page* ??.)

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}
```

(*End definition for* \showoverfull*. This function is documented on page* ??.)

\showoutput
\loggingoutput

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

(*End definition for* \showoutput *and* \loggingoutput*. These functions are documented on page* ??.)

\tracingall
\loggingall

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\loggingall}{etex tracing}%
⟨*2ekernel | latexrelease⟩
\ifx\tracingscantokens\@undefined
\gdef\loggingall{%
  \tracingstats\tw@
  \tracingpages\@ne
  \tracinglostchars\@ne
  \tracingparagraphs\@ne
  \errorcontextlines\maxdimen
  \loggingoutput
  \tracingmacros\tw@
  \tracingcommands\tw@
  \tracingrestores\@ne
  }%
\else
\gdef\loggingall{%
  \tracingstats\tw@
  \tracingpages\@ne
```

```
  \tracinglostchars\tw@
  \tracingparagraphs\@ne
  \tracinggroups\@ne
  \tracingifs\@ne
  \tracingscantokens\@ne
  \tracingnesting\@ne
  \errorcontextlines\maxdimen
  \loggingoutput
  \tracingmacros\tw@
  \tracingcommands\thr@@
  \tracingrestores\@ne
  \tracingassigns\@ne
}%
\fi
\gdef\tracingall{\showoverfull\loggingall}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\loggingall}{etex tracing}%
⟨latexrelease⟩\gdef\loggingall{\tracingcommands\tw@\tracingstats\tw@
⟨latexrelease⟩   \tracingpages\@ne\tracinglostchars\@ne
⟨latexrelease⟩   \tracingmacros\tw@\tracingparagraphs\@ne\tracingrestores\@ne
⟨latexrelease⟩   \errorcontextlines\maxdimen\loggingoutput}
⟨latexrelease⟩   \gdef\tracingall{\loggingall\showoverfull}
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\tracingall` *and* `\loggingall`*. These functions are documented on page* **??**.)

`\tracingnone`
`\hideoutput`

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\tracingnone}%
⟨latexrelease⟩                              {turn off etex tracing}%
⟨*2ekernel | latexrelease⟩
\ifx\tracingscantokens\@undefined
\def\tracingnone{%
  \tracingonline\z@
  \tracingcommands\z@
  \showboxdepth\m@ne
  \showboxbreadth\m@ne
  \tracingoutput\z@
  \errorcontextlines\m@ne
  \tracingrestores\z@
  \tracingparagraphs\z@
  \tracingmacros\z@
  \tracinglostchars\@ne
  \tracingpages\z@
  \tracingstats\z@
}%
\else
\def\tracingnone{%
  \tracingassigns\z@
  \tracingrestores\z@
  \tracingonline\z@
  \tracingcommands\z@
  \showboxdepth\m@ne
  \showboxbreadth\m@ne
```

```
  \tracingoutput\z@
  \errorcontextlines\m@ne
  \tracingnesting\z@
  \tracingscantokens\z@
  \tracingifs\z@
  \tracinggroups\z@
  \tracingparagraphs\z@
  \tracingmacros\z@
  \tracinglostchars\@ne
  \tracingpages\z@
  \tracingstats\z@
}%
\fi
```

```
\def\hideoutput{%
  \tracingoutput\z@
  \showboxbreadth\m@ne
  \showboxdepth\m@ne
  \tracingonline\m@ne
}%
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\tracingnone}%
⟨latexrelease⟩                              {turn off etex tracing}%
⟨latexrelease⟩\let\tracingnone\@undefined
⟨latexrelease⟩\let\hideoutput\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\tracingnone` *and* `\hideoutput`*. These functions are documented on page* **??**.)

LaTeX change: `\showhyphens` Defined later.

Punctuation affects the spacing.

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

File c
ltvers.dtx

1 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`
`\latexreleaseversion`
`\patch@level`

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

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

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

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

```
⟨*2ekernel⟩
\def\fmtname{LaTeX2e}
\edef\fmtversion
⟨/2ekernel⟩
⟨latexrelease⟩\edef\latexreleaseversion
⟨*2ekernel | latexrelease⟩
   {2020-10-01}
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
\def\patch@level{2}
```

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

`\development@branch@name`

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

(*End definition for* `\fmtname` *and others. These functions are documented on page* **??**.)

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%
!!!
```

```
⟨latexrelease⟩\newif\if@includeinrelease
⟨latexrelease⟩\@includeinreleasefalse
```

```
\def\IncludeInRelease#1{%
 \if@includeinrelease
   \PackageError{latexrelease}{mis-matched IncludeInRelease}%
                 {There is an \string\EndIncludeRelease\space missing}%
  \@includeinreleasefalse
  \fi
  \kernel@ifnextchar[%
  {\@IncludeInRelease{#1}}
  {\@IncludeInRelease{#1}[#1]}}
```

If a specific date has not been specified in `latexrelease` use '`#1`'.

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

```
\def\@IncludeInRele@se#1#2#3{%
  \toks@{[#1] #3}%
  \expandafter\ifx\csname\string#2+\@currname+IIR\endcsname\relax
```

If we roll back and the first patch already match then applying that is actually reapplying what is already in the format, i.e., it is useless and possibly allocating new registers. However, it makes the logic simpler so this is the way it is for now. In theory we could always jump overthe first patch because that is only really needed for rolling forward. So maybe one day ...

```
    \ifnum\expandafter\@parse@version#1//00\@nil
          >\expandafter\@parse@version\fmtversion//00\@nil
      \GenericInfo{}{Skipping: \the\toks@}%
     \expandafter\expandafter\expandafter\@gobble@IncludeInRelease
    \else
      \GenericInfo{}{Applying: \the\toks@}%
      \@includeinreleasetrue
      \expandafter\let\csname\string#2+\@currname+IIR\endcsname\@empty
    \fi
  \else
    \GenericInfo{}{Already applied: \the\toks@}%
    \expandafter\@gobble@IncludeInRelease
  \fi
}
```

```
\def\EndIncludeInRelease{%
\if@includeinrelease
  \@includeinreleasefalse
\else
  \PackageError{latexrelease}{mis-matched EndIncludeInRelease}{}%
\fi}
```

```
\long\def\@gobble@IncludeInRelease#1\EndIncludeInRelease{%
  \@includeinreleasefalse
  \@check@IncludeInRelease#1\IncludeInRelease\@check@IncludeInRelease
  \@end@check@IncludeInRelease}
```

```
\long\def\@check@IncludeInRelease#1\IncludeInRelease
                                  #2#3\@end@check@IncludeInRelease{%
  \ifx\@check@IncludeInRelease#2\else
    \PackageError{latexrelease}{skipped IncludeInRelease for tag \string#2}{}%
  \fi}
```

112 ⟨/2ekernel | latexrelease⟩

(*End definition for* `\IncludeInRelease`*. This function is documented on page* **??**.)

File d
ltluatex.dtx

1 Overview

LuaTEX adds a number of engine-specific functions to TEX. Several of these require set up that is best done in the kernel or need related support functions. This file provides *basic* support for LuaTEX at the LATEX 2_ε kernel level plus as a loadable file which can be used with plain TEX and LATEX.

This file contains code for both TEX (to be stored as part of the format) and Lua (to be loaded at the start of each job). In the Lua code, the kernel uses the namespace `luatexbase`.

The following `\count` registers are used here for register allocation:

`\e@alloc@attribute@count` Attributes (default 258)

`\e@alloc@ccodetable@count` Category code tables (default 259)

`\e@alloc@luafunction@count` Lua functions (default 260)

`\e@alloc@whatsit@count` User whatsits (default 261)

`\e@alloc@bytecode@count` Lua bytecodes (default 262)

`\e@alloc@luachunk@count` Lua chunks (default 263)

(`\count 256` is used for `\newmarks` allocation and `\count 257` is used for `\newXeTeXintercharclass` with XeTEX, with code defined in `ltfinal.dtx`). With any LATEX 2_ε kernel from 2015 onward these registers are part of the block in the extended area reserved by the kernel (prior to 2015 the LATEX 2_ε kernel did not provide any functionality for the extended allocation area).

2 Core TEX functionality

The commands defined here are defined for possible inclusion in a future LATEX format, however also extracted to the file `ltluatex.tex` which may be used with older LATEX formats, and with plain TEX.

`\newattribute` `\newattribute{`⟨*attribute*⟩`}`
Defines a named `\attribute`, indexed from 1 (*i.e.* `\attribute0` is never defined). Attributes initially have the marker value `-"7FFFFFFF` ('unset') set by the engine.

`\newcatcodetable` `\newcatcodetable{`⟨*catcodetable*⟩`}`
Defines a named `\catcodetable`, indexed from 1 (`\catcodetable0` is never assigned). A new catcode table will be populated with exactly those values assigned by IniTEX (as described in the LuaTEX manual).

`\newluafunction` `\newluafunction{`⟨*function*⟩`}`
Defines a named `\luafunction`, indexed from 1. (Lua indexes tables from 1 so `\luafunction0` is not available).

`\newwhatsit` `\newwhatsit{`⟨*whatsit*⟩`}`
Defines a custom `\whatsit`, indexed from 1.

`\newluabytecode` `\newluabytecode{`⟨*bytecode*⟩`}`

Allocates a number for Lua bytecode register, indexed from 1.

\newluachunkname `newluachunkname{⟨chunkname⟩}`
Allocates a number for Lua chunk register, indexed from 1. Also enters the name of the regiser (without backslash) into the `lua.name` table to be used in stack traces.

\catcodetable@initex
\catcodetable@string
\catcodetable@latex
\catcodetable@atletter
\setattribute
\unsetattribute

Predefined category code tables with the obvious assignments. Note that the `latex` and `atletter` tables set the full Unicode range to the codes predefined by the kernel.

`\setattribute{⟨attribute⟩}{⟨value⟩}`
`\unsetattribute{⟨attribute⟩}`
Set and unset attributes in a manner analogous to `\setlength`. Note that attributes take a marker value when unset so this operation is distinct from setting the value to zero.

3 Plain TEX interface

The ltluatex interface may be used with plain TEX using `\input{ltluatex}`. This inputs `ltluatex.tex` which inputs `etex.src` (or `etex.sty` if used with LATEX) if it is not already input, and then defines some internal commands to allow the ltluatex interface to be defined.

The luatexbase package interface may also be used in plain TEX, as before, by inputting the package `\input luatexbase.sty`. The new version of luatexbase is based on this ltluatex code but implements a compatibility layer providing the interface of the original package.

4 Lua functionality

4.1 Allocators in Lua

new_attribute `luatexbase.new_attribute(⟨attribute⟩)`
Returns an allocation number for the ⟨*attribute*⟩, indexed from 1. The attribute will be initialised with the marker value `-"7FFFFFFF` ('unset'). The attribute allocation sequence is shared with the TEX code but this function does *not* define a token using `\attributedef`. The attribute name is recorded in the `attributes` table. A metatable is provided so that the table syntax can be used consistently for attributes declared in TEX or Lua.

new_whatsit `luatexbase.new_whatsit(⟨whatsit⟩)`
Returns an allocation number for the custom ⟨*whatsit*⟩, indexed from 1.

new_bytecode `luatexbase.new_bytecode(⟨bytecode⟩)`
Returns an allocation number for a bytecode register, indexed from 1. The optional ⟨*name*⟩ argument is just used for logging.

new_chunkname `luatexbase.new_chunkname(⟨chunkname⟩)`
Returns an allocation number for a Lua chunk name for use with `\directlua` and `\latelua`, indexed from 1. The number is returned and also ⟨*name*⟩ argument is added to the `lua.name` array at that index.

new_luafunction `luatexbase.new_luafunction(⟨functionname⟩)`
Returns an allocation number for a lua function for use with `\luafunction`, `\lateluafunction`, and `\luadef`, indexed from 1. The optional ⟨*functionname*⟩ argument is just used for logging.

These functions all require access to a named TEX count register to manage their allocations. The standard names are those defined above for access from TEX,

e.g. \e@alloc@attribute@count, but these can be adjusted by defining the variable ⟨*type*⟩`_count_name` before loading `ltluatex.lua`, for example

```
local attribute_count_name = "attributetracker"
require("ltluatex")
```

would use a TEX `\count` (`\countdef`'d token) called `attributetracker` in place of \e@alloc@attribute@count.

4.2 Lua access to TEX register numbers

`registernumber` `luatexbase.registernumer(`⟨*name*⟩`)`

Sometimes (notably in the case of Lua attributes) it is necessary to access a register *by number* that has been allocated by TEX. This package provides a function to look up the relevant number using LuaTEX's internal tables. After for example `\newattribute\myattrib`, `\myattrib` would be defined by (say) `\myattrib=\attribute15`. `luatexbase.registernumer("myattrib")` would then return the register number, 15 in this case. If the string passed as argument does not correspond to a token defined by `\attributedef`, `\countdef` or similar commands, the Lua value `false` is returned.

As an example, consider the input:

```
\newcommand\test[1]{%
\typeout{#1: \expandafter\meaning\csname#1\endcsname^^J
\space\space\space\space
\directlua{tex.write(luatexbase.registernumber("#1") or "bad input")}%
}}

\test{undefinedrubbish}

\test{space}

\test{hbox}

\test{@MM}

\test{@tempdima}
\test{@tempdimb}

\test{strutbox}

\test{sixt@@n}

\attrbutedef\myattr=12
\myattr=200
\test{myattr}
```

If the demonstration code is processed with LuaLATEX then the following would be produced in the log and terminal output.

```
undefinedrubbish: \relax
     bad input
```

```
space: macro:->
     bad input
hbox: \hbox
     bad input
@MM: \mathchar"4E20
     20000
@tempdima: \dimen14
     14
@tempdimb: \dimen15
     15
strutbox: \char"B
     11
sixt@@n: \char"10
     16
myattr: \attribute12
     12
```

Notice how undefined commands, or commands unrelated to registers do not produce an error, just return `false` and so print `bad input` here. Note also that commands defined by `\newbox` work and return the number of the box register even though the actual command holding this number is a `\chardef` defined token (there is no `\boxdef`).

4.3 Module utilities

`provides_module` `luatexbase.provides_module(`⟨*info*⟩`)`
This function is used by modules to identify themselves; the `info` should be a table containing information about the module. The required field `name` must contain the name of the module. It is recommended to provide a field `date` in the usual LaTeX format `yyyy/mm/dd`. Optional fields `version` (a string) and `description` may be used if present. This information will be recorded in the log. Other fields are ignored.

`module_info` `luatexbase.module_info(`⟨*module*⟩, ⟨*text*⟩`)`
`module_warning` `luatexbase.module_warning(`⟨*module*⟩, ⟨*text*⟩`)`
`module_error` `luatexbase.module_error(`⟨*module*⟩, ⟨*text*⟩`)`
These functions are similar to LaTeX's `\PackageError`, `\PackageWarning` and `\PackageInfo` in the way they format the output. No automatic line breaking is done, you may still use `\n` as usual for that, and the name of the package will be prepended to each output line.

Note that `luatexbase.module_error` raises an actual Lua error with `error()`, which currently means a call stack will be dumped. While this may not look pretty, at least it provides useful information for tracking the error down.

4.4 Callback management

`add_to_callback` `luatexbase.add_to_callback(`⟨*callback*⟩, ⟨*function*⟩, ⟨*description*⟩`)` Registers the ⟨*function*⟩ into the ⟨*callback*⟩ with a textual ⟨*description*⟩ of the function. Functions are inserted into the callback in the order loaded.

`remove_from_callback` `luatexbase.remove_from_callback(`⟨*callback*⟩, ⟨*description*⟩`)` Removes the callback function with ⟨*description*⟩ from the ⟨*callback*⟩. The removed function and its description are returned as the results of this function.

`in_callback` `luatexbase.in_callback(`⟨*callback*⟩, ⟨*description*⟩`)` Checks if the ⟨*description*⟩ matches one of the functions added to the list for the ⟨*callback*⟩, returning a boolean value.

`disable_callback` `luatexbase.disable_callback(`⟨*callback*⟩`)` Sets the ⟨*callback*⟩ to `false` as described in the LuaTEX manual for the underlying `callback.register` built-in. Callbacks will only be set to false (and thus be skipped entirely) if there are no functions registered using the callback.

`callback_descriptions` A list of the descriptions of functions registered to the specified callback is returned. `{}` is returned if there are no functions registered.

`create_callback` `luatexbase.create_callback(`⟨*name*⟩,metatype,⟨*default*⟩`)` Defines a user defined callback. The last argument is a default function or `false`.

`call_callback` `luatexbase.call_callback(`⟨*name*⟩,...`)` Calls a user defined callback with the supplied arguments.

5 Implementation

```
⟨*2ekernel | tex | latexrelease⟩
⟨2ekernel | latexrelease⟩\ifx\directlua\@undefined\else
```

5.1 Minimum LuaTEX version

LuaTEX has changed a lot over time. In the kernel support for ancient versions is not provided: trying to build a format with a very old binary therefore gives some information in the log and loading stops. The cut-off selected here relates to the tree-searching behaviour of `require()`: from version 0.60, LuaTEX will correctly find Lua files in the `texmf` tree without 'help'.

```
⟨latexrelease⟩\IncludeInRelease{2015/10/01}
⟨latexrelease⟩                  {\newluafunction}{LuaTeX}%
\ifnum\luatexversion<60 %
  \wlog{****************************************************}
  \wlog{* LuaTeX version too old for ltluatex support *}
  \wlog{****************************************************}
  \expandafter\endinput
\fi
```

Two simple LaTeX macros from `ltdefns.dtx` have to be defined here because ltdefns.dtx is not loaded yet when ltluatex.dtx is executed.

```
\long\def\@gobble#1{}
\long\def\@firstofone#1{#1}
```

5.2 Older LaTeX/Plain TeX setup

```
⟨*tex⟩
```

Older LaTeX formats don't have the primitives with 'native' names: sort that out. If they already exist this will still be safe.

```
\directlua{tex.enableprimitives("",tex.extraprimitives("luatex"))}
\ifx\e@alloc\@undefined
```

In pre-2014 LaTeX, or plain TeX, load `etex.{sty,src}`.

```
  \ifx\documentclass\@undefined
    \ifx\loccount\@undefined
      \input{etex.src}%
    \fi
    \catcode`\@=11 %
    \outer\expandafter\def\csname newfam\endcsname
```

```
                          {\alloc@8\fam\chardef\et@xmaxfam}
  \else
    \RequirePackage{etex}
    \expandafter\def\csname newfam\endcsname
                    {\alloc@8\fam\chardef\et@xmaxfam}
    \expandafter\let\expandafter\new@mathgroup\csname newfam\endcsname
  \fi
```

5.2.1 Fixes to etex.src/etex.sty

These could and probably should be made directly in an update to `etex.src` which already has some LuaTEX-specific code, but does not define the correct range for LuaTEX.

2015-07-13 higher range in luatex.

```
\edef \et@xmaxregs {\ifx\directlua\@undefined 32768\else 65536\fi}
```

luatex/xetex also allow more math fam.

```
\edef \et@xmaxfam {\ifx\Umathcode\@undefined\sixt@@n\else\@cclvi\fi}
```

```
\count 270=\et@xmaxregs % locally allocates \count registers
\count 271=\et@xmaxregs % ditto for \dimen registers
\count 272=\et@xmaxregs % ditto for \skip registers
\count 273=\et@xmaxregs % ditto for \muskip registers
\count 274=\et@xmaxregs % ditto for \box registers
\count 275=\et@xmaxregs % ditto for \toks registers
\count 276=\et@xmaxregs % ditto for \marks classes
```

and 256 or 16 fam. (Done above due to plain/LaTeX differences in ltluatex.)

```
% \outer\def\newfam{\alloc@8\fam\chardef\et@xmaxfam}
```

End of proposed changes to `etex.src`

5.2.2 luatex specific settings

Switch to global cf `luatex.sty` to leave room for inserts not really needed for luatex but possibly most compatible with existing use.

```
\expandafter\let\csname newcount\expandafter\expandafter\endcsname
                \csname globcount\endcsname
\expandafter\let\csname newdimen\expandafter\expandafter\endcsname
                \csname globdimen\endcsname
\expandafter\let\csname newskip\expandafter\expandafter\endcsname
                \csname globskip\endcsname
\expandafter\let\csname newbox\expandafter\expandafter\endcsname
                \csname globbox\endcsname
```

Define`\e@alloc` as in latex (the existing macros in `etex.src` hard to extend to further register types as they assume specific 26x and 27x count range. For compatibility the existing register allocation is not changed.

```
\chardef\e@alloc@top=65535
\let\e@alloc@chardef\chardef
```

```
\def\e@alloc#1#2#3#4#5#6{%
  \global\advance#3\@ne
  \e@ch@ck{#3}{#4}{#5}#1%
  \allocationnumber#3\relax
  \global#2#6\allocationnumber
  \wlog{\string#6=\string#1\the\allocationnumber}}%
```

```
\gdef\e@ch@ck#1#2#3#4{%
  \ifnum#1<#2\else
    \ifnum#1=#2\relax
      #1\@cclvi
      \ifx\count#4\advance#1 10 \fi
    \fi
    \ifnum#1<#3\relax
    \else
      \errmessage{No room for a new \string#4}%
    \fi
  \fi}%
```

Fix up allocations not to clash with `etex.src`.

```
\expandafter\csname newcount\endcsname\e@alloc@attribute@count
\expandafter\csname newcount\endcsname\e@alloc@ccodetable@count
\expandafter\csname newcount\endcsname\e@alloc@luafunction@count
\expandafter\csname newcount\endcsname\e@alloc@whatsit@count
\expandafter\csname newcount\endcsname\e@alloc@bytecode@count
\expandafter\csname newcount\endcsname\e@alloc@luachunk@count
```

End of conditional setup for plain TEX / old LATEX.

```
\fi
```
⟨/tex⟩

5.3 Attributes

`\newattribute` As is generally the case for the LuaTEX registers we start here from 1. Notably, some code assumes that `\attribute0` is never used so this is important in this case.

```
\ifx\e@alloc@attribute@count\@undefined
  \countdef\e@alloc@attribute@count=258
  \e@alloc@attribute@count=\z@
\fi
\def\newattribute#1{%
  \e@alloc\attribute\attributedef
    \e@alloc@attribute@count\m@ne\e@alloc@top#1%
}
```

(*End definition for* `\newattribute`*. This function is documented on page* **??**.)

`\setattribute` `\unsetattribute` Handy utilities.

```
\def\setattribute#1#2{#1=\numexpr#2\relax}
\def\unsetattribute#1{#1=-"7FFFFFFF\relax}
```

(*End definition for* `\setattribute` *and* `\unsetattribute`*. These functions are documented on page* **??**.)

5.4 Category code tables

`\newcatcodetable` Category code tables are allocated with a limit half of that used by LuaTEX for everything else. At the end of allocation there needs to be an initialisation step. Table 0 is already taken (it's the global one for current use) so the allocation starts at 1.

```
\ifx\e@alloc@ccodetable@count\@undefined
  \countdef\e@alloc@ccodetable@count=259
  \e@alloc@ccodetable@count=\z@
\fi
```

```
\def\newcatcodetable#1{%
  \e@alloc\catcodetable\chardef
    \e@alloc@ccodetable@count\m@ne{"8000}#1%
  \initcatcodetable\allocationnumber
}
```

(*End definition for* `\newcatcodetable`*. This function is documented on page* **??**.)

`\catcodetable@initex`
`\catcodetable@string`
`\catcodetable@latex`
`\catcodetable@atletter`

Save a small set of standard tables. The Unicode data is read here in using a parser simplified from that in `load-unicode-data`: only the nature of letters needs to be detected.

```
\newcatcodetable\catcodetable@initex
\newcatcodetable\catcodetable@string
\begingroup
  \def\setrangecatcode#1#2#3{%
    \ifnum#1>#2 %
      \expandafter\@gobble
    \else
      \expandafter\@firstofone
    \fi
      {%
        \catcode#1=#3 %
        \expandafter\setrangecatcode\expandafter
          {\number\numexpr#1 + 1\relax}{#2}{#3}
      }%
  }
  \@firstofone{%
    \catcodetable\catcodetable@initex
      \catcode0=12 %
      \catcode13=12 %
      \catcode37=12 %
      \setrangecatcode{65}{90}{12}%
      \setrangecatcode{97}{122}{12}%
      \catcode92=12 %
      \catcode127=12 %
      \savecatcodetable\catcodetable@string
    \endgroup
  }%
\newcatcodetable\catcodetable@latex
\newcatcodetable\catcodetable@atletter
\begingroup
  \def\parseunicodedataI#1;#2;#3;#4\relax{%
    \parseunicodedataII#1;#3;#2 First>\relax
  }%
  \def\parseunicodedataII#1;#2;#3 First>#4\relax{%
    \ifx\relax#4\relax
      \expandafter\parseunicodedataIII
    \else
      \expandafter\parseunicodedataIV
    \fi
      {#1}#2\relax%
  }%
  \def\parseunicodedataIII#1#2#3\relax{%
    \ifnum 0%
      \if L#21\fi
```

```
      \if M#21\fi
      >0 %
      \catcode"#1=11 %
    \fi
  }%
  \def\parseunicodedataIV#1#2#3\relax{%
    \read\unicoderead to \unicodedataline
    \if L#2%
      \count0="#1 %
      \expandafter\parseunicodedataV\unicodedataline\relax
    \fi
  }%
  \def\parseunicodedataV#1;#2\relax{%
    \loop
      \unless\ifnum\count0>"#1 %
        \catcode\count0=11 %
        \advance\count0 by 1 %
    \repeat
  }%
  \def\storedpar{\par}%
  \chardef\unicoderead=\numexpr\count16 + 1\relax
  \openin\unicoderead=UnicodeData.txt %
  \loop\unless\ifeof\unicoderead %
    \read\unicoderead to \unicodedataline
    \unless\ifx\unicodedataline\storedpar
      \expandafter\parseunicodedataI\unicodedataline\relax
    \fi
  \repeat
  \closein\unicoderead
  \@firstofone{%
    \catcode64=12 %
    \savecatcodetable\catcodetable@latex
    \catcode64=11 %
    \savecatcodetable\catcodetable@atletter
   }
\endgroup
```

(*End definition for* `\catcodetable@initex` *and others. These functions are documented on page* **??**.)

5.5 Named Lua functions

`\newluafunction` Much the same story for allocating LuaTEX functions except here they are just numbers so they are allocated in the same way as boxes. Lua indexes from 1 so once again slot 0 is skipped.

```
\ifx\e@alloc@luafunction@count\@undefined
  \countdef\e@alloc@luafunction@count=260
  \e@alloc@luafunction@count=\z@
\fi
\def\newluafunction{%
  \e@alloc\luafunction\e@alloc@chardef
    \e@alloc@luafunction@count\m@ne\e@alloc@top
}
```

(*End definition for* `\newluafunction`. *This function is documented on page* **??**.)

5.6 Custom whatsits

\newwhatsit These are only settable from Lua but for consistency are definable here.

```
\ifx\e@alloc@whatsit@count\@undefined
  \countdef\e@alloc@whatsit@count=261
  \e@alloc@whatsit@count=\z@
\fi
\def\newwhatsit#1{%
  \e@alloc\whatsit\e@alloc@chardef
    \e@alloc@whatsit@count\m@ne\e@alloc@top#1%
}
```

(*End definition for* \newwhatsit*. This function is documented on page* ??.)

5.7 Lua bytecode registers

\newluabytecode These are only settable from Lua but for consistency are definable here.

```
\ifx\e@alloc@bytecode@count\@undefined
  \countdef\e@alloc@bytecode@count=262
  \e@alloc@bytecode@count=\z@
\fi
\def\newluabytecode#1{%
  \e@alloc\luabytecode\e@alloc@chardef
    \e@alloc@bytecode@count\m@ne\e@alloc@top#1%
}
```

(*End definition for* \newluabytecode*. This function is documented on page* ??.)

5.8 Lua chunk registers

\newluachunkname As for bytecode registers, but in addition we need to add a string to the `lua.name` table to use in stack tracing. We use the name of the command passed to the allocator, with no backslash.

```
\ifx\e@alloc@luachunk@count\@undefined
  \countdef\e@alloc@luachunk@count=263
  \e@alloc@luachunk@count=\z@
\fi
\def\newluachunkname#1{%
  \e@alloc\luachunk\e@alloc@chardef
    \e@alloc@luachunk@count\m@ne\e@alloc@top#1%
    {\escapechar\m@ne
    \directlua{lua.name[\the\allocationnumber]="\string#1"}}%
}
```

(*End definition for* \newluachunkname*. This function is documented on page* ??.)

5.9 Lua loader

Lua code loaded in the format often has to to be loaded again at the beginning of every job, so we define a helper whch allows us to avoid duplicated code:

```
\def\now@and@everyjob#1{%
  \everyjob\expandafter{\the\everyjob
    #1%
```

```
  }%
  #1%
}
```

Load the Lua code at the start of every job. For the conversion of TEX into numbers at the Lua side we need some known registers: for convenience we use a set of systematic names, which means using a group around the Lua loader.

```
⟨2ekernel⟩\now@and@everyjob{%
  \begingroup
    \attributedef\attributezero=0 %
    \chardef     \charzero     =0 %
```

Note name change required on older luatex, for hash table access.

```
    \countdef    \CountZero    =0 %
    \dimendef    \dimenzero    =0 %
    \mathchardef \mathcharzero =0 %
    \muskipdef   \muskipzero   =0 %
    \skipdef     \skipzero     =0 %
    \toksdef     \tokszero     =0 %
    \directlua{require("ltluatex")}
  \endgroup
⟨2ekernel⟩}
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}
⟨latexrelease⟩                  {\newluafunction}{LuaTeX}%
⟨latexrelease⟩\let\e@alloc@attribute@count\@undefined
⟨latexrelease⟩\let\newattribute\@undefined
⟨latexrelease⟩\let\setattribute\@undefined
⟨latexrelease⟩\let\unsetattribute\@undefined
⟨latexrelease⟩\let\e@alloc@ccodetable@count\@undefined
⟨latexrelease⟩\let\newcatcodetable\@undefined
⟨latexrelease⟩\let\catcodetable@initex\@undefined
⟨latexrelease⟩\let\catcodetable@string\@undefined
⟨latexrelease⟩\let\catcodetable@latex\@undefined
⟨latexrelease⟩\let\catcodetable@atletter\@undefined
⟨latexrelease⟩\let\e@alloc@luafunction@count\@undefined
⟨latexrelease⟩\let\newluafunction\@undefined
⟨latexrelease⟩\let\e@alloc@luafunction@count\@undefined
⟨latexrelease⟩\let\newwhatsit\@undefined
⟨latexrelease⟩\let\e@alloc@whatsit@count\@undefined
⟨latexrelease⟩\let\newluabytecode\@undefined
⟨latexrelease⟩\let\e@alloc@bytecode@count\@undefined
⟨latexrelease⟩\let\newluachunkname\@undefined
⟨latexrelease⟩\let\e@alloc@luachunk@count\@undefined
⟨latexrelease⟩\directlua{luatexbase.uninstall()}
⟨latexrelease⟩\EndIncludeInRelease
```

In `\everyjob`, if luaotfload is available, load it and switch to TU.

```
⟨latexrelease⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease⟩                  {\fontencoding}{TU in everyjob}%
⟨latexrelease⟩\fontencoding{TU}\let\encodingdefault\f@encoding
⟨latexrelease⟩\ifx\directlua\@undefined\else
⟨2ekernel⟩\everyjob\expandafter{%
⟨2ekernel⟩   \the\everyjob
```

```
⟨*2ekernel, latexrelease⟩
  \directlua{%
  if xpcall(function ()%
              require('luaotfload-main')%
             end,texio.write_nl) then %
  local _void = luaotfload.main ()%
  else %
  texio.write_nl('Error in luaotfload: reverting to OT1')%
  tex.print('\string\\def\string\\encodingdefault{OT1}')%
  end %
  }%
  \let\f@encoding\encodingdefault
  \expandafter\let\csname ver@luaotfload.sty\endcsname\fmtversion
⟨/2ekernel, latexrelease⟩
⟨latexrelease⟩\fi
⟨2ekernel⟩  }
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\fontencoding}{TU in everyjob}%
⟨latexrelease⟩\fontencoding{OT1}\let\encodingdefault\f@encoding
⟨latexrelease⟩\EndIncludeInRelease

⟨2ekernel | latexrelease⟩\fi
⟨/2ekernel | tex | latexrelease⟩
```

5.10 Lua module preliminaries

```
⟨*lua⟩
```

Some set up for the Lua module which is needed for all of the Lua functionality added here.

`luatexbase` Set up the table for the returned functions. This is used to expose all of the public functions.

```
luatexbase       = luatexbase or { }
local luatexbase = luatexbase
```

(*End definition for* `luatexbase`*. This function is documented on page* ??.)

Some Lua best practice: use local versions of functions where possible.

```
local string_gsub      = string.gsub
local tex_count        = tex.count
local tex_setattribute = tex.setattribute
local tex_setcount     = tex.setcount
local texio_write_nl   = texio.write_nl

local luatexbase_warning
local luatexbase_error
```

5.11 Lua module utilities

5.11.1 Module tracking

`modules` To allow tracking of module usage, a structure is provided to store information and to return it.

```
local modules = modules or { }
```

(*End definition for* `modules`*. This function is documented on page* ??.)

`provides_module` Local function to write to the log.

```
local function luatexbase_log(text)
  texio_write_nl("log", text)
end
```

Modelled on `\ProvidesPackage`, we store much the same information but with a little more structure.

```
local function provides_module(info)
  if not (info and info.name) then
    luatexbase_error("Missing module name for provides_module")
  end
  local function spaced(text)
    return text and (" " .. text) or ""
  end
  luatexbase_log(
    "Lua module: " .. info.name
      .. spaced(info.date)
      .. spaced(info.version)
      .. spaced(info.description)
  )
  modules[info.name] = info
end
luatexbase.provides_module = provides_module
```

(*End definition for* `provides_module`*. This function is documented on page* ??.)

5.11.2 Module messages

There are various warnings and errors that need to be given. For warnings we can get exactly the same formatting as from TEX. For errors we have to make some changes. Here we give the text of the error in the LATEX format then force an error from Lua to halt the run. Splitting the message text is done using `\n` which takes the place of `\MessageBreak`.

First an auxiliary for the formatting: this measures up the message leader so we always get the correct indent.

```
local function msg_format(mod, msg_type, text)
  local leader = ""
  local cont
  local first_head
  if mod == "LaTeX" then
    cont = string_gsub(leader, ".", " ")
    first_head = leader .. "LaTeX: "
  else
    first_head = leader .. "Module "  .. msg_type
    cont = "(" .. mod .. ")"
      .. string_gsub(first_head, ".", " ")
    first_head =  leader .. "Module "  .. mod .. " " .. msg_type  .. ":"
  end
  if msg_type == "Error" then
    first_head = "\n" .. first_head
  end
  if string.sub(text,-1) ~= "\n" then
```

```
    text = text .. " "
  end
  return first_head .. " "
    .. string_gsub(
         text
 .. "on input line "
         .. tex.inputlineno, "\n", "\n" .. cont .. " "
      )
   .. "\n"
end
```

`module_info`
`module_warning`
`module_error`

Write messages.

```
local function module_info(mod, text)
  texio_write_nl("log", msg_format(mod, "Info", text))
end
luatexbase.module_info = module_info
local function module_warning(mod, text)
  texio_write_nl("term and log",msg_format(mod, "Warning", text))
end
luatexbase.module_warning = module_warning
local function module_error(mod, text)
  error(msg_format(mod, "Error", text))
end
luatexbase.module_error = module_error
```

(*End definition for* `module_info`*,* `module_warning`*, and* `module_error`*. These functions are documented on page* ??.)

Dedicated versions for the rest of the code here.

```
function luatexbase_warning(text)
  module_warning("luatexbase", text)
end
function luatexbase_error(text)
  module_error("luatexbase", text)
end
```

5.12 Accessing register numbers from Lua

Collect up the data from the TEX level into a Lua table: from version 0.80, LuaTEX makes that easy.

```
local luaregisterbasetable = { }
local registermap = {
  attributezero = "assign_attr"    ,
  charzero      = "char_given"     ,
  CountZero     = "assign_int"     ,
  dimenzero     = "assign_dimen"   ,
  mathcharzero  = "math_given"     ,
  muskipzero    = "assign_mu_skip" ,
  skipzero      = "assign_skip"    ,
  tokszero      = "assign_toks"    ,
}
local createtoken
if tex.luatexversion > 81 then
  createtoken = token.create
elseif tex.luatexversion > 79 then
```

```
  createtoken = newtoken.create
end
local hashtokens    = tex.hashtokens()
local luatexversion = tex.luatexversion
for i,j in pairs (registermap) do
  if luatexversion < 80 then
    luaregisterbasetable[hashtokens[i][1]] =
      hashtokens[i][2]
  else
    luaregisterbasetable[j] = createtoken(i).mode
  end
end
```

`registernumber` Working out the correct return value can be done in two ways. For older LuaTEX releases it has to be extracted from the `hashtokens`. On the other hand, newer LuaTEX's have `newtoken`, and whilst `.mode` isn't currently documented, Hans Hagen pointed to this approach so we should be OK.

```
local registernumber
if luatexversion < 80 then
  function registernumber(name)
    local nt = hashtokens[name]
    if(nt and luaregisterbasetable[nt[1]]) then
      return nt[2] - luaregisterbasetable[nt[1]]
    else
      return false
    end
  end
else
  function registernumber(name)
    local nt = createtoken(name)
    if(luaregisterbasetable[nt.cmdname]) then
      return nt.mode - luaregisterbasetable[nt.cmdname]
    else
      return false
    end
  end
end
luatexbase.registernumber = registernumber
```

(*End definition for* `registernumber`*. This function is documented on page* **??**.)

5.13 Attribute allocation

`new_attribute` As attributes are used for Lua manipulations its useful to be able to assign from this end.

```
local attributes=setmetatable(
{},
{
__index = function(t,key)
return registernumber(key) or nil
end}
)
luatexbase.attributes = attributes
```

```
local attribute_count_name =
                     attribute_count_name or "e@alloc@attribute@count"
local function new_attribute(name)
  tex_setcount("global", attribute_count_name,
                         tex_count[attribute_count_name] + 1)
  if tex_count[attribute_count_name] > 65534 then
    luatexbase_error("No room for a new \\attribute")
  end
  attributes[name]= tex_count[attribute_count_name]
  luatexbase_log("Lua-only attribute " .. name .. " = " ..
                 tex_count[attribute_count_name])
  return tex_count[attribute_count_name]
end
luatexbase.new_attribute = new_attribute
```

(*End definition for* `new_attribute`*. This function is documented on page* **??**.)

5.14 Custom whatsit allocation

`new_whatsit` Much the same as for attribute allocation in Lua.

```
local whatsit_count_name = whatsit_count_name or "e@alloc@whatsit@count"
local function new_whatsit(name)
  tex_setcount("global", whatsit_count_name,
                         tex_count[whatsit_count_name] + 1)
  if tex_count[whatsit_count_name] > 65534 then
    luatexbase_error("No room for a new custom whatsit")
  end
  luatexbase_log("Custom whatsit " .. (name or "") .. " = " ..
                 tex_count[whatsit_count_name])
  return tex_count[whatsit_count_name]
end
luatexbase.new_whatsit = new_whatsit
```

(*End definition for* `new_whatsit`*. This function is documented on page* **??**.)

5.15 Bytecode register allocation

`new_bytecode` Much the same as for attribute allocation in Lua. The optional ⟨*name*⟩ argument is used in the log if given.

```
local bytecode_count_name =
                         bytecode_count_name or "e@alloc@bytecode@count"
local function new_bytecode(name)
  tex_setcount("global", bytecode_count_name,
                         tex_count[bytecode_count_name] + 1)
  if tex_count[bytecode_count_name] > 65534 then
    luatexbase_error("No room for a new bytecode register")
  end
  luatexbase_log("Lua bytecode " .. (name or "") .. " = " ..
                 tex_count[bytecode_count_name])
  return tex_count[bytecode_count_name]
end
luatexbase.new_bytecode = new_bytecode
```

(*End definition for* `new_bytecode`*. This function is documented on page* **??**.)

5.16 Lua chunk name allocation

`new_chunkname` As for bytecode registers but also store the name in the `lua.name` table.

```
local chunkname_count_name =
                        chunkname_count_name or "e@alloc@luachunk@count"
local function new_chunkname(name)
  tex_setcount("global", chunkname_count_name,
                         tex_count[chunkname_count_name] + 1)
  local chunkname_count = tex_count[chunkname_count_name]
  chunkname_count = chunkname_count + 1
  if chunkname_count > 65534 then
    luatexbase_error("No room for a new chunkname")
  end
  lua.name[chunkname_count]=name
  luatexbase_log("Lua chunkname " .. (name or "") .. " = " ..
                 chunkname_count .. "\n")
  return chunkname_count
end
luatexbase.new_chunkname = new_chunkname
```

(*End definition for* `new_chunkname`*. This function is documented on page* **??**.)

5.17 Lua function allocation

`new_luafunction` Much the same as for attribute allocation in Lua. The optional ⟨*name*⟩ argument is used in the log if given.

```
local luafunction_count_name =
                         luafunction_count_name or "e@alloc@luafunction@count"
local function new_luafunction(name)
  tex_setcount("global", luafunction_count_name,
                         tex_count[luafunction_count_name] + 1)
  if tex_count[luafunction_count_name] > 65534 then
    luatexbase_error("No room for a new luafunction register")
  end
  luatexbase_log("Lua function " .. (name or "") .. " = " ..
                 tex_count[luafunction_count_name])
  return tex_count[luafunction_count_name]
end
luatexbase.new_luafunction = new_luafunction
```

(*End definition for* `new_luafunction`*. This function is documented on page* **??**.)

5.18 Lua callback management

The native mechanism for callbacks in LuaTEX allows only one per function. That is extremely restrictive and so a mechanism is needed to add and remove callbacks from the appropriate hooks.

5.18.1 Housekeeping

The main table: keys are callback names, and values are the associated lists of functions. More precisely, the entries in the list are tables holding the actual function as `func` and

the identifying description as `description`. Only callbacks with a non-empty list of functions have an entry in this list.

```
local callbacklist = callbacklist or { }
```

Numerical codes for callback types, and name-to-value association (the table keys are strings, the values are numbers).

```
local list, data, exclusive, simple, reverselist = 1, 2, 3, 4, 5
local types   = {
  list        = list,
  data        = data,
  exclusive   = exclusive,
  simple      = simple,
  reverselist = reverselist,
}
```

Now, list all predefined callbacks with their current type, based on the LuaTEX manual version 1.01. A full list of the currently-available callbacks can be obtained using

```
\directlua{
  for i,_ in pairs(callback.list()) do
    texio.write_nl("- " .. i)
  end
}
\bye
```

in plain LuaTEX. (Some undocumented callbacks are omitted as they are to be removed.)

```
local callbacktypes = callbacktypes or {
```

Section 8.2: file discovery callbacks.

```
  find_read_file     = exclusive,
  find_write_file    = exclusive,
  find_font_file     = data,
  find_output_file   = data,
  find_format_file   = data,
  find_vf_file       = data,
  find_map_file      = data,
  find_enc_file      = data,
  find_pk_file       = data,
  find_data_file     = data,
  find_opentype_file = data,
  find_truetype_file = data,
  find_type1_file    = data,
  find_image_file    = data,

  open_read_file     = exclusive,
  read_font_file     = exclusive,
  read_vf_file       = exclusive,
  read_map_file      = exclusive,
  read_enc_file      = exclusive,
  read_pk_file       = exclusive,
  read_data_file     = exclusive,
  read_truetype_file = exclusive,
  read_type1_file    = exclusive,
  read_opentype_file = exclusive,
```

Not currently used by luatex but included for completeness. may be used by a font handler.

```
find_cidmap_file   = data,
read_cidmap_file   = exclusive,
```

Section 8.3: data processing callbacks.

```
process_input_buffer  = data,
process_output_buffer = data,
process_jobname       = data,
```

Section 8.4: node list processing callbacks.

```
contribute_filter      = simple,
buildpage_filter       = simple,
build_page_insert      = exclusive,
pre_linebreak_filter   = list,
linebreak_filter       = exclusive,
append_to_vlist_filter = exclusive,
post_linebreak_filter  = reverselist,
hpack_filter           = list,
vpack_filter           = list,
hpack_quality          = list,
vpack_quality          = list,
pre_output_filter      = list,
process_rule           = exclusive,
hyphenate              = simple,
ligaturing             = simple,
kerning                = simple,
insert_local_par       = simple,
pre_mlist_to_hlist_filter = list,
mlist_to_hlist         = exclusive,
post_mlist_to_hlist_filter = reverselist,
new_graf               = exclusive,
```

Section 8.5: information reporting callbacks.

```
pre_dump             = simple,
start_run            = simple,
stop_run             = simple,
start_page_number    = simple,
stop_page_number     = simple,
show_error_hook      = simple,
show_warning_message = simple,
show_error_message   = simple,
show_lua_error_hook  = simple,
start_file           = simple,
stop_file            = simple,
call_edit            = simple,
finish_synctex       = simple,
wrapup_run           = simple,
```

Section 8.6: PDF-related callbacks.

```
finish_pdffile            = data,
finish_pdfpage            = data,
page_objnum_provider      = data,
page_order_index          = data,
process_pdf_image_content = data,
```

Section 8.7: font-related callbacks.

```
  define_font                     = exclusive,
  glyph_info                      = exclusive,
  glyph_not_found                 = exclusive,
  glyph_stream_provider           = exclusive,
  make_extensible                 = exclusive,
  font_descriptor_objnum_provider = exclusive,

}
luatexbase.callbacktypes=callbacktypes
```

`callback.register` Save the original function for registering callbacks and prevent the original being used. The original is saved in a place that remains available so other more sophisticated code can override the approach taken by the kernel if desired.

```
local callback_register = callback_register or callback.register
function callback.register()
  luatexbase_error("Attempt to use callback.register() directly\n")
end
```

(*End definition for* `callback.register`*. This function is documented on page* **??**.)

5.18.2 Handlers

The handler function is registered into the callback when the first function is added to this callback's list. Then, when the callback is called, the handler takes care of running all functions in the list. When the last function is removed from the callback's list, the handler is unregistered.

More precisely, the functions below are used to generate a specialized function (closure) for a given callback, which is the actual handler.

The way the functions are combined together depends on the type of the callback. There are currently 4 types of callback, depending on the calling convention of the functions the callback can hold:

simple is for functions that don't return anything: they are called in order, all with the same argument;

data is for functions receiving a piece of data of any type except node list head (and possibly other arguments) and returning it (possibly modified): the functions are called in order, and each is passed the return value of the previous (and the other arguments untouched, if any). The return value is that of the last function;

list is a specialized variant of *data* for functions filtering node lists. Such functions may return either the head of a modified node list, or the boolean values `true` or `false`. The functions are chained the same way as for *data* except that for the following. If one function returns `false`, then `false` is immediately returned and the following functions are *not* called. If one function returns `true`, then the same head is passed to the next function. If all functions return `true`, then `true` is returned, otherwise the return value of the last function not returning `true` is used.

reverselist is a specialized variant of *list* which executes functions in inverse order.

exclusive is for functions with more complex signatures; functions in this type of callback are *not* combined: An error is raised if a second callback is registered..

Handler for `data` callbacks.

```
local function data_handler(name)
  return function(data, ...)
    for _,i in ipairs(callbacklist[name]) do
      data = i.func(data,...)
    end
    return data
  end
end
```

Default for user-defined `data` callbacks without explicit default.

```
local function data_handler_default(value)
  return value
end
```

Handler for `exclusive` callbacks. We can assume `callbacklist[name]` is not empty: otherwise, the function wouldn't be registered in the callback any more.

```
local function exclusive_handler(name)
  return function(...)
    return callbacklist[name][1].func(...)
  end
end
```

Handler for `list` callbacks.

```
local function list_handler(name)
  return function(head, ...)
    local ret
    local alltrue = true
    for _,i in ipairs(callbacklist[name]) do
      ret = i.func(head, ...)
      if ret == false then
        luatexbase_warning(
          "Function ‘" .. i.description .. "’ returned false\n"
            .. "in callback ‘" .. name .."’"
         )
        return false
      end
      if ret ~= true then
        alltrue = false
        head = ret
      end
    end
    return alltrue and true or head
  end
end
```

Default for user-defined `list` and `reverselist` callbacks without explicit default.

```
local function list_handler_default()
  return true
end
```

Handler for `reverselist` callbacks.

```
local function reverselist_handler(name)
  return function(head, ...)
    local ret
    local alltrue = true
```

```
    local callbacks = callbacklist[name]
    for i = #callbacks, 1, -1 do
      local cb = callbacks[i]
      ret = cb.func(head, ...)
      if ret == false then
        luatexbase_warning(
          "Function ‘" .. cb.description .. "’ returned false\n"
            .. "in callback ‘" .. name .."’"
         )
        return false
      end
      if ret ~= true then
        alltrue = false
        head = ret
      end
    end
    return alltrue and true or head
  end
end
```

Handler for `simple` callbacks.

```
local function simple_handler(name)
  return function(...)
    for _,i in ipairs(callbacklist[name]) do
      i.func(...)
    end
  end
end
```

Default for user-defined `simple` callbacks without explicit default.

```
local function simple_handler_default()
end
```

Keep a handlers table for indexed access and a table with the corresponding default functions.

```
local handlers  = {
  [data]        = data_handler,
  [exclusive]   = exclusive_handler,
  [list]        = list_handler,
  [reverselist] = reverselist_handler,
  [simple]      = simple_handler,
}
local defaults = {
  [data]        = data_handler_default,
  [exclusive]   = nil,
  [list]        = list_handler_default,
  [reverselist] = list_handler_default,
  [simple]      = simple_handler_default,
}
```

5.18.3 Public functions for callback management

Defining user callbacks perhaps should be in package code, but impacts on `add_to_callback`. If a default function is not required, it may be declared as `false`. First we need a list of user callbacks.

```
local user_callbacks_defaults = {
  pre_mlist_to_hlist_filter = list_handler_default,
  mlist_to_hlist = node.mlist_to_hlist,
  post_mlist_to_hlist_filter = list_handler_default,
}
```

`create_callback` The allocator itself.

```
local function create_callback(name, ctype, default)
  local ctype_id = types[ctype]
  if not name  or name  == ""
  or not ctype_id
  then
    luatexbase_error("Unable to create callback:\n" ..
                     "valid callback name and type required")
  end
  if callbacktypes[name] then
    luatexbase_error("Unable to create callback `" .. name ..
                     "':\ncallback is already defined")
  end
  default = default or defaults[ctype_id]
  if not default then
    luatexbase_error("Unable to create callback `" .. name ..
                     "':\ndefault is required for `" .. ctype ..
                     "' callbacks")
  elseif type (default) ~= "function" then
    luatexbase_error("Unable to create callback `" .. name ..
                     "':\ndefault is not a function")
  end
  user_callbacks_defaults[name] = default
  callbacktypes[name] = ctype_id
end
luatexbase.create_callback = create_callback
```

(*End definition for* `create_callback`*. This function is documented on page* **??**.)

`call_callback` Call a user defined callback. First check arguments.

```
local function call_callback(name,...)
  if not name or name == "" then
    luatexbase_error("Unable to create callback:\n" ..
                     "valid callback name required")
  end
  if user_callbacks_defaults[name] == nil then
    luatexbase_error("Unable to call callback `" .. name
                     .. "':\nunknown or empty")
   end
  local l = callbacklist[name]
  local f
  if not l then
    f = user_callbacks_defaults[name]
  else
    f = handlers[callbacktypes[name]](name)
  end
  return f(...)
end
luatexbase.call_callback=call_callback
```

(*End definition for* `call_callback`. *This function is documented on page* **??**.)

`add_to_callback` Add a function to a callback. First check arguments.

```
local function add_to_callback(name, func, description)
  if not name or name == "" then
    luatexbase_error("Unable to register callback:\n" ..
                     "valid callback name required")
  end
  if not callbacktypes[name] or
    type(func) ~= "function" or
    not description or
    description == "" then
    luatexbase_error(
      "Unable to register callback.\n\n"
        .. "Correct usage:\n"
        .. "add_to_callback(<callback>, <function>, <description>)"
    )
  end
```

Then test if this callback is already in use. If not, initialise its list and register the proper handler.

```
  local l = callbacklist[name]
  if l == nil then
    l = { }
    callbacklist[name] = l
```

If it is not a user defined callback use the primitive callback register.

```
    if user_callbacks_defaults[name] == nil then
      callback_register(name, handlers[callbacktypes[name]](name))
    end
  end
```

Actually register the function and give an error if more than one `exclusive` one is registered.

```
  local f = {
    func        = func,
    description = description,
  }
  local priority = #l + 1
  if callbacktypes[name] == exclusive then
    if #l == 1 then
      luatexbase_error(
        "Cannot add second callback to exclusive function\n‘" ..
        name .. "’")
    end
  end
  table.insert(l, priority, f)
```

Keep user informed.

```
  luatexbase_log(
    "Inserting ‘" .. description .. "’ at position "
      .. priority .. " in ‘" .. name .. "’."
  )
end
luatexbase.add_to_callback = add_to_callback
```

(*End definition for* `add_to_callback`. *This function is documented on page* **??**.)

`remove_from_callback` Remove a function from a callback. First check arguments.

```
local function remove_from_callback(name, description)
  if not name or name == "" then
    luatexbase_error("Unable to remove function from callback:\n" ..
                     "valid callback name required")
  end
  if not callbacktypes[name] or
    not description or
    description == "" then
    luatexbase_error(
      "Unable to remove function from callback.\n\n"
        .. "Correct usage:\n"
        .. "remove_from_callback(<callback>, <description>)"
    )
  end
  local l = callbacklist[name]
  if not l then
    luatexbase_error(
      "No callback list for ‘" .. name .. "’\n")
  end
```

Loop over the callback's function list until we find a matching entry. Remove it and check if the list is empty: if so, unregister the callback handler.

```
  local index = false
  for i,j in ipairs(l) do
    if j.description == description then
      index = i
      break
    end
  end
  if not index then
    luatexbase_error(
      "No callback ‘" .. description .. "’ registered for ‘" ..
      name .. "’\n")
  end
  local cb = l[index]
  table.remove(l, index)
  luatexbase_log(
    "Removing  ‘" .. description .. "’ from ‘" .. name .. "’."
  )
  if #l == 0 then
    callbacklist[name] = nil
    if user_callbacks_defaults[name] == nil then
      callback_register(name, nil)
    end
  end
  return cb.func,cb.description
end
luatexbase.remove_from_callback = remove_from_callback
```

(*End definition for* `remove_from_callback`. *This function is documented on page* **??**.)

`in_callback` Look for a function description in a callback.

```
local function in_callback(name, description)
  if not name
    or name == ""
    or not callbacklist[name]
    or not callbacktypes[name]
    or not description then
      return false
  end
  for _, i in pairs(callbacklist[name]) do
    if i.description == description then
      return true
    end
  end
  return false
end
luatexbase.in_callback = in_callback
```

(*End definition for* `in_callback`*. This function is documented on page* ??.)

`disable_callback` As we subvert the engine interface we need to provide a way to access this functionality.

```
local function disable_callback(name)
  if(callbacklist[name] == nil) then
    callback_register(name, false)
  else
    luatexbase_error("Callback list for " .. name .. " not empty")
  end
end
luatexbase.disable_callback = disable_callback
```

(*End definition for* `disable_callback`*. This function is documented on page* ??.)

`callback_descriptions` List the descriptions of functions registered for the given callback.

```
local function callback_descriptions (name)
  local d = {}
  if not name
    or name == ""
    or not callbacklist[name]
    or not callbacktypes[name]
    then
    return d
  else
  for k, i in pairs(callbacklist[name]) do
    d[k]= i.description
    end
  end
  return d
end
luatexbase.callback_descriptions =callback_descriptions
```

(*End definition for* `callback_descriptions`*. This function is documented on page* ??.)

`uninstall` Unlike at the TEX level, we have to provide a back-out mechanism here at the same time as the rest of the code. This is not meant for use by anything other than latexrelease: as such this is *deliberately* not documented for users!

```
local function uninstall()
  module_info(
    "luatexbase",
    "Uninstalling kernel luatexbase code"
  )
  callback.register = callback_register
  luatexbase = nil
end
luatexbase.uninstall = uninstall
```

(*End definition for* `uninstall`*. This function is documented on page* **??**.)

`mlist_to_hlist` To emulate these callbacks, the "real" `mlist_to_hlist` is replaced by a wrapper calling the wrappers before and after.

```
callback_register("mlist_to_hlist", function(head, display_type, need_penalties)
  local current = call_callback("pre_mlist_to_hlist_filter", head, display_type, need_penalti
  if current == false then
    flush_list(head)
    return nil
  elseif current == true then
    current = head
  end
  current = call_callback("mlist_to_hlist", current, display_type, need_penalties)
  local post = call_callback("post_mlist_to_hlist_filter", current, display_type, need_penalt
  if post == true then
    return current
  elseif post == false then
    flush_list(current)
    return nil
  end
  return post
end)
```

(*End definition for* `mlist_to_hlist`*. This function is documented on page* **??**.)

```
⟨/lua⟩
```

Reset the catcode of `@`.

```
⟨tex⟩\catcode`\@=\etatcatcode\relax
```

File e
ltexpl.dtx

1 expl3-dependent code

1.1 Loader

First define some blank commands, so that in case something goes wrong while loading expl3, we won't get strange `Undefined control sequence` errors.

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩            {\@expl@sys@load@backend@@}{Roll forward support}%
\def\@expl@sys@load@backend@@{}
\def\@expl@push@filename@@{}
\def\@expl@push@filename@aux@@{}
\def\@expl@pop@filename@@{}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
```

Create a hook for last-minute expl3 material.

```
⟨*2ekernel⟩
\def\@expl@finalise@setup@@{}
⟨/2ekernel⟩
```

Now define some basics to support loading expl3. These macros can be defined here safely, because they are redefined later on by the kernel, so we define simpler versions just to suit our needs.

```
⟨*2ekernel⟩
\long\def\@gobble#1{}
\long\def\@firstofone#1{#1}
\long\def\@firstoftwo#1#2{#1}
\long\def\@secondoftwo#1#2{#2}
\long\def\IfFileExists#1{%
  \openin\@inputcheck"#1" %
  \ifeof\@inputcheck
    \expandafter\@secondoftwo
  \else
    \closein\@inputcheck
    \expandafter\@firstoftwo
  \fi}
\long\def\@ifnextchar#1#2#3{%
  \let\reserved@d=#1%
  \def\reserved@a{#2}%
  \def\reserved@b{#3}%
  \futurelet\@let@token\@ifnch}
\def\@ifnch{%
  \ifx\@let@token\reserved@d
    \expandafter\reserved@a
  \else
    \expandafter\reserved@b
  \fi}
⟨/2ekernel⟩
```

If we are doing a rollback with a format containing expl3 we aren't reloading it as that creates havoc. This may need a refined version!

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {expl3}{Pre-load expl3}%
\expandafter\ifx\csname tex\string _let:D\endcsname\relax
  \expandafter\@firstofone
\else
  \GenericInfo{}{Skipping: expl3 code already part of the format}%
⟨2ekernel⟩   \expandafter\endinput
⟨latexrelease⟩   \expandafter\@gobble
\fi
```

Check for the required primitive/engine support and the existence of a loader.

```
  {%
    \IfFileExists{expl3.ltx}
      {%
        \ifnum0%
          \ifdefined\pdffilesize 1\fi
          \ifdefined\filesize 1\fi
          \ifdefined\luatexversion\ifnum\luatexversion>94 1\fi\fi
          \ifdefined\kanjiskip 1\fi
            >0 %
          \expandafter\@firstofone
        \else
```

In `2ekernel` mode, an error is fatal and building the format is aborted. Use `\batchmode \read -1 to \tokenlist`, which errors with `! Emergency stop. (cannot \read from t` and aborts the TEX run. In `latexrelease` mode, raise an error and do nothing. Both ways, the error message shows the minimum expl3 engine requirements.

```
⟨2ekernel⟩          \def~{ }\def\MessageBreak{^^J~~~~~~~~~~~~~~~}%
⟨2ekernel⟩          \errmessage{LaTeX Error:
⟨latexrelease⟩          \@latex@error{%
            LaTeX requires the e-TeX primitives and additional\MessageBreak
            functionality available in the engines:\MessageBreak
              - pdfTeX v1.40\MessageBreak
              - XeTeX v0.99992\MessageBreak
              - LuaTeX v0.95\MessageBreak
              - e-(u)pTeX mid-2012\MessageBreak
            or later%
⟨latexrelease⟩          }\@ehd \expandafter\@gobble
⟨2ekernel⟩          }\batchmode \read -1 to \reserved@a
        \fi
      }
      {%
⟨*2ekernel⟩
        \errmessage{LaTeX requires expl3}%
        \batchmode \read -1 to \reserved@a
⟨/2ekernel⟩
```

We do not support a roll forward across 2019. You need to start with 2019 if you want to get to 2020 or beyond.

```
⟨*latexrelease⟩
        \@latex@warning@no@line
          {You need a format that already contains a recent\MessageBreak
```

```
            expl3 as part of the kernel, e.g. at least a kernel\MessageBreak
            from 2019 to roll forward to that date!\MessageBreak
            --- I'm giving up!\MessageBreak\MessageBreak
            Note that manually loading the expl3 package\MessageBreak
            from your distribution is not enough}%
          \@@end  % die
⟨/latexrelease⟩
       }%
       {%
         \input expl3.ltx
         \ifdefined\NewDocumentCommand
         \else
           \IfFileExists{xparse.ltx}
             {\input xparse.ltx }
             {}%
          \fi
       }%
  }
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                 {expl3}{Pre-load expl3}%
⟨latexrelease⟩\IfFileExists{expl3.ltx}
⟨latexrelease⟩  {%
⟨latexrelease⟩    \ifnum0%
⟨latexrelease⟩      \ifdefined\pdffilesize 1\fi
⟨latexrelease⟩      \ifdefined\filesize 1\fi
⟨latexrelease⟩      \ifdefined\luatexversion\ifnum\luatexversion>94 1\fi\fi
⟨latexrelease⟩      >0 %
⟨latexrelease⟩    \else
⟨latexrelease⟩      \message{Skipping expl3-dependent extensions}
⟨latexrelease⟩      \expandafter\@gobbletwo
⟨latexrelease⟩    \fi
⟨latexrelease⟩  }
⟨latexrelease⟩  {%
⟨latexrelease⟩    \message{Skipping expl3-dependent extensions}%
⟨latexrelease⟩    \@gobbletwo
⟨latexrelease⟩  }%
⟨latexrelease⟩\input{expl3.ltx}
⟨latexrelease⟩\EndIncludeInRelease
```

Starting with 2020-10 the kernel contains most of the `xparse` interfaces loaded above. These interfaces are later used to define various user command. However, if somebody with an earlier installation tries to roll forward (using a `latexrelease` package from a newer release) this will not work as the earlier kernel doesn't contain commands such as `\NewDocumentCommand`.

We therefore add code into `latexrelease.sty` to load the `xparse` package of the current installation instead. this may not work if the installation is really old, but roll forward can't do miracles.

If the kernel is 2020-10 or later `\NewDocumentCommand` is already known (whether or not you roll forward or back) so nothing happens, but if you start from am earlier kernel the `xparse` package gets loaded by `latexrelease`.

```
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\NewDocumentCommand}{Emergency loading xparse}%
```

```
⟨latexrelease⟩
⟨latexrelease⟩\ifdefined\NewDocumentCommand \else
⟨latexrelease⟩   \RequirePackage{xparse}
⟨latexrelease⟩\fi
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

1.2 Using expl3 code

In order to ease the implemantation of some new features in LaTeX 2_ε we may (temporarily) use some coding based on the expl3-code. Such macros will eventually vanish and may be changed unannounced. They are there for internal use in the LaTeX 2_ε kernel and are not meant to be used in third-party packages. These macros will always have the `@expl@` prefix in their name.

The rest of the name matches the expl3 name but with all underscores replaced by `@`s and the `:` replaced by `@@`, e.g.,

```
\cs_new_eq:NN \@expl@tl@trim@spaces@apply@@nN \tl_trim_spaces_apply:nN
```

if that expl3 command is needed in places that are others coded in LaTeX 2_ε conventions.

In this file, each release of LaTeX adds an `\IncludeInRelease` block, in which the macros copied for that release were defined. In case a rollback is requested, the entire block is changed.

Each macro copied has a `\changes` entry to explain when and why it was copied, so that further to that may spot it easily.

Here `\cs_gset_eq:NN` is used, instead of the `new` variant because if different releases use that same name for different purposes, each can copy the macro without worrying about redefinitions.

```
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@expl@cs@to@str@@N}%
⟨latexrelease⟩        {expl3 macros added for the 2020-10-01 release}%
```

The expl3 activation needs to be inside the release guards as otherwise rolling forward is broken in old kernels that do not have expl3 loaded.

```
\ExplSyntaxOn

\cs_gset_eq:NN \@expl@cs@to@str@@N \cs_to_str:N
\cs_gset_eq:NN \@expl@str@if@eq@@nnTF \str_if_eq:nnTF

\cs_gset_eq:NN \@expl@cs@prefix@spec@@N \cs_prefix_spec:N
\cs_gset_eq:NN \@expl@cs@argument@spec@@N \cs_argument_spec:N
\cs_gset_eq:NN \@expl@cs@replacement@spec@@N \cs_replacement_spec:N

\cs_gset_eq:NN \@expl@str@map@function@@NN \str_map_function:NN
\cs_gset_eq:NN \@expl@char@generate@@nn \char_generate:nn

\ExplSyntaxOff
```

Here we can't assume that expl3 is available. It will be if we roll back but if this code is executed rolling forward it needs to be pure 2e.

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@expl@cs@to@str@@N}%
⟨latexrelease⟩        {expl3 macros added for the 2020-10-01 release}%
⟨latexrelease⟩\let \@expl@cs@to@str@@N \@undefined
⟨latexrelease⟩\let \@expl@str@if@eq@@nnTF \@undefined
```

```
⟨latexrelease⟩\let \@expl@cs@prefix@spec@@N \@undefined
⟨latexrelease⟩\let \@expl@cs@argument@spec@@N \@undefined
⟨latexrelease⟩\let \@expl@cs@replacement@spec@@N \@undefined
⟨latexrelease⟩\let \@expl@str@map@function@@NN \@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
```

File f
ltdefns.dtx

1 Definitions

This section contains commands used in defining other macros.

```
⟨*2ekernel⟩
```

1.1 Initex initialisations

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

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

(*End definition for* `\two@digits`*. This function is documented on page* **??**.)

`\typeout` Display something on the terminal.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\typeout}{Allow "par" in \typeout}%
\protected\long\def\typeout#1{\begingroup
  \set@display@protect
  \def\par{^^J^^J}%
  \immediate\write\@unused{#1}\endgroup}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\typeout}{Allow "par" in \typeout}%
⟨latexrelease⟩
⟨latexrelease⟩\def\typeout#1{\begingroup\set@display@protect
⟨latexrelease⟩    \immediate\write\@unused{#1}\endgroup}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\typeout`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\newlinechar`*. This function is documented on page* **??**.)

1.2 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        %%%
```

(*End definition for* `\@@par`*. This function is documented on page* **??**.)

\@@hyph Save original primitive definition.

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

(*End definition for* `\@@hyph`. *This function is documented on page* **??**.)

\@@italiccorr Save the original italic correction.

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

(*End definition for* `\@@italiccorr`. *This function is documented on page* **??**.)

\@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}
```

The next one is another 100 tokens worth.

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

(*End definition for* `\@height` *and others. These functions are documented on page* **??**.)

```
\message{hacks,}
```

\hb@xt@

1.3 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}
```

```
\ifx\directlua\@undefined
```

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

```
\else
```

```
\def\@xtypein[#1]#2{%
  \typeout{#2}%
  \begingroup \endlinechar\m@ne
  \read\@inputcheck to#1%
  \expandafter\endgroup
  \expandafter\def\expandafter#1\expandafter{#1}%
  \@typein}%
```

```
\fi
```

(*End definition for* \typein*. This function is documented on page* ??.)

\@namedef

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

(*End definition for* \@namedef*. This function is documented on page* ??.)

\@nameuse

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

(*End definition for* \@nameuse*. This function is documented on page* ??.)

\@cons

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

(*End definition for* `\@cons`*. This function is documented on page* **??**.)

\@car
\@cdr

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

(*End definition for* `\@car` *and* `\@cdr`*. These functions are documented on page* **??**.)

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

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@carcube}{Make \@carcube long}%
⟨*2ekernel | latexrelease⟩
\long\def\@carcube#1#2#3#4\@nil{#1#2#3}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@carcube}{Undo: Make \@carcube long}%
⟨latexrelease⟩\def\@carcube#1#2#3#4\@nil{#1#2#3}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@carcube`*. This function is documented on page* **??**.)

\@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
```

(*End definition for* `\@onlypreamble` *and* `\@preamblecmds`*. These functions are documented on page* **??**.)

\@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}}
```

(*End definition for* `\@star@or@long`*. This function is documented on page* **??**.)

\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
```

(*End definition for* `\l@ngrel@x`*. This function is documented on page* **??**.)

\newcommand

User level `\newcommand`.

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

\new@command

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

(*End definition for* `\newcommand` *and* `\new@command`*. These functions are documented on page* **??**.)

\@newcommand
\@argdef
\@xargdef

Handling arguments for `\newcommand`.

```
\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 existing command names. "Default" is the contents of the second optional argument of `(re)newcommand`.

```
     \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}}}
```

(*End definition for* `\@newcommand`*,* `\@argdef`*, and* `\@xargdef`*. These functions are documented on page* **??**.)

\@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}]}}
```

(*End definition for* `\@testopt`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* \@protected@testopt. *This function is documented on page* ??.)

\@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%
}
```

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

(*End definition for* \@yargdef *and* \@yargd@f. *These functions are documented on page* ??.)

\@reargdef

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

(*End definition for* `\@reargdef`*. This function is documented on page* **??**.)

`\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{Command \string#1 undefined}\@ehc}%
     \relax
  \let\@ifdefinable\@rc@ifdefinable
  \new@command#1}
```

(*End definition for* `\renewcommand` *and* `\renew@command`*. These functions are documented on page* **??**.)

`\@ifdefinable` `\@@ifdefinable` `\@rc@ifdefinable` Test if 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}
```

(*End definition for* `\@ifdefinable`*,* `\@@ifdefinable`*, and* `\@rc@ifdefinable`*. These functions are documented on page* **??**.)

`\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}]}}
```

(*End definition for* `\newenvironment` *and others. These functions are documented on page* **??**.)

\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
  \expandafter\let\csname end#1\endcsname\relax
  \new@environment{#1}}
```

(*End definition for* `\renewenvironment` *and* `\renew@environment`*. These functions are documented on page* **??**.)

\@newenv

The internal version of `\newenvironment`.

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

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

```
\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}%
     \l@ngrel@x\expandafter\def\csname end#1\endcsname{#4}}
```

(*End definition for* `\@newenv`*. This function is documented on page* **??**.)

\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
    \let#1\iffalse
    \@if#1\iftrue
    \@if#1\iffalse
  \escapechar\count@}
```

\@if

```
\def\@if#1#2{%
  \expandafter\def\csname\expandafter\@gobbletwo\string#1%
                     \expandafter\@gobbletwo\string#2\endcsname
                        {\let#1#2}}
```

(*End definition for* `\newif` *and* `\@if`*. These functions are documented on page* **??**.)

`\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}%
```

(*End definition for* `\providecommand` *and* `\provide@command`*. These functions are documented on page* **??**.)

`\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
```

(*End definition for* `\CheckCommand` *and* `\check@command`*. These functions are documented on page* **??**.)

`\@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
```

(*End definition for* `\@check@c`*. This function is documented on page* **??**.)

\@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
```

(*End definition for* \@check@eq*. This function is documented on page* ??.)

\@gobble \@gobbletwo \@gobblethree \@gobblefour The \@gobble macro is used to get rid of its argument.

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

(*End definition for* \@gobble *and others. These functions are documented on page* ??.)

\@firstofone \@firstoftwo \@secondoftwo Some argument-grabbers.

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

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

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

\@iden (*End definition for* \@firstofone *and others. These functions are documented on page* ??.)

\@thirdofthree Another grabber now used in the encoding specific section.

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

(*End definition for* \@thirdofthree*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@expandtwoargs*. This function is documented on page* ??.)

\@backslashchar A category code 12 backslash.

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

(*End definition for* \@backslashchar*. This function is documented on page* ??.)

1.4 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 occasions 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 four 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. More precisely, whenever the content of an `\edef` or `\xdef` etc. can contain arbitrary user input not under the direct control of the programmer, one should use `\proetected@edef` instead of `\edef`, etc., so that `\protect` has a suitable definition and the user input will not break if it contains fragile commands.

`\@unexpandable@protect`

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

(*End definition for* `\@unexpandable@protect`. *This function is documented on page* **??**.)

`\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 \protect to \@empty will cause _ to loop forever. It will also break lots of other things, such as protected \ifmmodes inside \halignns. If you really have to do such a thing, then please set \@typeset@protect to be \@empty as well. (This is what the code for \patterns does, for example.)

More fun with \expandafter and \csname.

```
\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}%
   \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
}
```

(*End definition for* \DeclareRobustCommand *and* \declare@robustcommand*. These functions are documented on page* ??.)

\@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%
}
```

(*End definition for* \@x@protect *and* \x@protect*. These functions are documented on page* ??.)

\@typeset@protect

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

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

(*End definition for* `\@typeset@protect`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\set@display@protect` *and* `\set@typeset@protect`*. These functions are documented on page* **??**.)

`\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}
```

(*End definition for* `\protected@edef` *and others. These functions are documented on page* **??**.)

`\protect`

The normal meaning of `\protect`

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

(*End definition for* `\protect`*. This function is documented on page* **??**.)

`\MakeRobust`

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

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

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\MakeRobust}{\MakeRobust}%
⟨*2ekernel | latexrelease⟩
\def\MakeRobust#1{%
  \count@=\escapechar
```

```
  \escapechar=`\\
  \@ifundefined{\expandafter\@gobble\string#1}{%
    \@latex@error{The control sequence `\string#1' is undefined!%
      \MessageBreak There is nothing here to make robust}%
    \@eha
  }%
```

Then we check if the macro is already robust. We do this by testing if the internal name for a robust macro is defined, namely `\foo␣`. If it is already defined do nothing, otherwise set `\foo␣` equal to `\foo` and redefine `\foo` so that it acts like a macro defined with `\DeclareRobustCommand`. We use `\@kernel@rename@newcommand` to copy `\foo` over to `\foo␣`, including a possible default optional argument.

```
  {%
    \@ifundefined{\expandafter\@gobble\string#1\space}%
    {%
      \expandafter\@kernel@rename@newcommand
        \csname\expandafter\@gobble\string#1\space\endcsname
        #1%
      \edef\reserved@a{\string#1}%
      \def\reserved@b{#1}%
      \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
      \xdef#1{%
        \ifx\reserved@a\reserved@b
          \noexpand\x@protect\noexpand#1%
        \fi
        \noexpand\protect\expandafter\noexpand
        \csname\expandafter\@gobble\string#1\space\endcsname}%
    }%
    {\@latex@info{The control sequence `\string#1' is already robust}}%
  }%
  \escapechar=\count@
}%
```

`\@kernel@rename@newcommand`

This macro renames a command, possibly with an optional argument (defined with `\newcommand`) from `#2` to `#1`, by renaming the internal macro `\\#2` to `\\#1` and defining `\#1` appropriately, then undefining `\#2` and `\\#2`. The `\afterassignment` trick is to make both definitions in `\@copy@newcommand` global (which are local by default).

In case the macro was defined with `\newcommand` and an optional argument, to replicate exactly the behaviour of `\DeclareRobustCommand` we have to move also the internal `\\foo` to `\\foo␣`. In that case, `#1` will be a parameterless macro (`\robust@command@chk@safe` checks that), and `\@if@newcommand` will return true (both defined below in this file). If so, we can use `\@copy@newcommand` rather than plain `\let` to copy the command over. `\@kernel@rename@newcommand` does this test and carries out the renaming.

```
\def\@kernel@rename@newcommand#1#2{%
  \robust@command@chk@safe#2%
    {\@if@newcommand#2%
      {\afterassignment\global
       \global\@copy@newcommand#1#2%
       \global\let#2\@undefined
       \global\expandafter\let\csname\string#2\endcsname\@undefined}%
      {\global\let#1=#2}}%
    {\global\let#1=#2}}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{2019/10/01}{\MakeRobust}{\MakeRobust}%
⟨latexrelease⟩\def\MakeRobust#1{%
⟨latexrelease⟩  \@ifundefined{\expandafter\@gobble\string#1}{%
⟨latexrelease⟩    \@latex@error{The control sequence `\string#1' is undefined!%
⟨latexrelease⟩      \MessageBreak There is nothing here to make robust}%
⟨latexrelease⟩    \@eha
⟨latexrelease⟩  }%
⟨latexrelease⟩  {%
⟨latexrelease⟩    \@ifundefined{\expandafter\@gobble\string#1\space}%
⟨latexrelease⟩    {%
⟨latexrelease⟩      \global\expandafter\let\csname
⟨latexrelease⟩      \expandafter\@gobble\string#1\space\endcsname=#1%
⟨latexrelease⟩      \edef\reserved@a{\string#1}%
⟨latexrelease⟩      \def\reserved@b{#1}%
⟨latexrelease⟩      \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
⟨latexrelease⟩      \xdef#1{%
⟨latexrelease⟩        \ifx\reserved@a\reserved@b
⟨latexrelease⟩          \noexpand\x@protect\noexpand#1%
⟨latexrelease⟩        \fi
⟨latexrelease⟩        \noexpand\protect\expandafter\noexpand
⟨latexrelease⟩        \csname\expandafter\@gobble\string#1\space\endcsname}%
⟨latexrelease⟩    }%
⟨latexrelease⟩    {\@latex@info{The control sequence `\string#1' is already robust}}%
⟨latexrelease⟩   }%
⟨latexrelease⟩}%
⟨latexrelease⟩\let\@kernel@rename@newcommand\@undefined
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\MakeRobust}{\MakeRobust}%
⟨latexrelease⟩\def\MakeRobust#1{%
⟨latexrelease⟩  \@ifundefined{\expandafter\@gobble\string#1}{%
⟨latexrelease⟩    \@latex@error{The control sequence `\string#1' is undefined!%
⟨latexrelease⟩      \MessageBreak There is nothing here to make robust}%
⟨latexrelease⟩    \@eha
⟨latexrelease⟩  }%
⟨latexrelease⟩  {%
⟨latexrelease⟩    \@ifundefined{\expandafter\@gobble\string#1\space}%
⟨latexrelease⟩    {%
⟨latexrelease⟩      \expandafter\let\csname
⟨latexrelease⟩      \expandafter\@gobble\string#1\space\endcsname=#1%
⟨latexrelease⟩      \edef\reserved@a{\string#1}%
⟨latexrelease⟩      \def\reserved@b{#1}%
⟨latexrelease⟩      \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
⟨latexrelease⟩      \edef#1{%
⟨latexrelease⟩        \ifx\reserved@a\reserved@b
⟨latexrelease⟩          \noexpand\x@protect\noexpand#1%
⟨latexrelease⟩        \fi
⟨latexrelease⟩        \noexpand\protect\expandafter\noexpand
⟨latexrelease⟩        \csname\expandafter\@gobble\string#1\space\endcsname}%
⟨latexrelease⟩    }%
⟨latexrelease⟩    {\@latex@info{The control sequence `\string#1' is already robust}}%
```

```
⟨latexrelease⟩    }%
⟨latexrelease⟩}%
⟨latexrelease⟩\let\@kernel@rename@newcommand\@undefined
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\MakeRobust}{\MakeRobust}%
⟨latexrelease⟩\let\MakeRobust\@undefined
⟨latexrelease⟩\let\@kernel@rename@newcommand\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\MakeRobust` *and* `\@kernel@rename@newcommand`*. These functions are documented on page* **??**.)

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

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\kernel@make@fragile}{Undo robustness}%
\def\kernel@make@fragile#1{%
  \@ifundefined{\expandafter\@gobble\string#1\space}%
```

If not robust do nothing.

```
      {}%
```

Otherwise copy `\foo␣` back to `\foo`. Then use `\@kernel@rename@newcommand` to check and copy `\\foo␣` back to `\\foo` in case the command has an optional argument. If so, also undefine `\\foo␣`, and at the end undefine `\foo␣`.

```
      {%
       \global\expandafter\let\expandafter #1\csname
         \expandafter\@gobble\string#1\space\endcsname
       \expandafter\@kernel@rename@newcommand
         \csname\expandafter\@gobble\string#1\expandafter\endcsname
         \csname\expandafter\@gobble\string#1\space\endcsname
       \global\expandafter\let\csname
         \expandafter\@gobble\string#1\space\endcsname\@undefined
      }%
}
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\kernel@make@fragile}{Undo robustness}%
⟨latexrelease⟩\def\kernel@make@fragile#1{%
⟨latexrelease⟩   \@ifundefined{\expandafter\@gobble\string#1\space}%
⟨latexrelease⟩      {}%
⟨latexrelease⟩      {%
⟨latexrelease⟩       \global\expandafter\let\expandafter #1\csname
⟨latexrelease⟩       \expandafter\@gobble\string#1\space\endcsname
⟨latexrelease⟩       \global\expandafter\let\csname
```

```
⟨latexrelease⟩        \expandafter\@gobble\string#1\space\endcsname\@undefined
⟨latexrelease⟩       }%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
```

(*End definition for* `\kernel@make@fragile`*. This function is documented on page* **??**.)

1.5 Acting on robust commands

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020-10-01}{\robust@command@act}
⟨latexrelease⟩   {Add \robust@command@act}%
⟨*2ekernel | latexrelease⟩
```

With most document level commands being robust now there is more of a requirement to have a standard way of aliasing (or copying) a command to a new name, for example to save an original definition before changing a command. `\DeclareCommandCopy` is analogous to TEX's `\let`, except that it copes with the different types of robust commands defined by LATEX's mechanisms.

A couple of "types of robustness" are defined by the LATEX 2_ε kernel, namely robust commands defined with `\DeclareRobustCommand` and commands with optional arguments defined with `\newcommand`. However there are other types of robust commands that are frequently used, which are not defined in the LATEX 2_ε kernel, like commands defined with xparse's `\NewDocumentCommand` and etoolbox's `\newrobustcmd`.

In this section we will define a generic extensible machinery to act on robust commands. This code will then be used to test if a command is robust, considered the different types of robustness, and then either copy that definition, if `\DeclareCommandCopy` (or similar) is used, or show the definition of the command, if `\ShowCommmand` is used.

`\robust@command@act` The looping machinery is generic and knows nothing about what is to be done for each case. The syntax of the main macro `\robust@command@act` is:

> `\robust@command@act`⟨*action-list*⟩⟨*robust-cmd*⟩
> ⟨*fallback-action*⟩⟨*act-arg*⟩

⟨*action-list*⟩ is a token list of the form:

> `{`⟨*if-type-1*⟩ ⟨*act-type-1*⟩`}`
> `{`⟨*if-type-2*⟩ ⟨*act-type-2*⟩`}`
> ...

`\robust@command@act` will iterate over the ⟨*action-list*⟩, evaluating each ⟨*if-type-n*⟩⟨*robust-cmd*⟩`{`⟨*true*⟩`}{` If the ⟨*if-type-n*⟩ conditional returns ⟨*true*⟩, then ⟨*act-type-n*⟩⟨*act-arg*⟩ is executed, and the loop ends. If the conditional returns ⟨*false*⟩, then ⟨*if-type-n* + 1⟩ is executed in the same way, until either one of the conditionals return ⟨*true*⟩, or the end of the ⟨*action-list*⟩ is reached. If the end is reached, then ⟨*fallback-action*⟩⟨*act-arg*⟩ is executed before `\robust@command@act` exits.

`\robust@command@act` will start by using `\robust@command@act@chk@args` to check if the ⟨*robust-cmd*⟩ (`#2`) is a parameterless (possibly `\protected`) macro. If it is not, the command is not a robust command: these always start with a parameterless user-level macro; in that case, `\robust@command@act@end` is used to short-circuit the process and do the ⟨*fallback-action*⟩ (`#3`). This first test is necessary because later on we need to be

able to expand the ⟨*robust-cmd*⟩ without the risk of it Breaking Badly, and as a bonus, this speeds up the process in case we used `\NewCommandCopy` in a "normal" macro.

```
\long\def\robust@command@act#1#2#3#4{%
  \robust@command@chk@safe#2%
    {\expandafter\robust@command@act@loop
       \expandafter#2%
         #1{\@nnil\@nnil}%
     \robust@command@act@end}%
    {\robust@command@act@end}%
      {#3}{#4}}%
```

`\robust@command@act@loop`
`\robust@command@act@loop@aux`
`\robust@command@act@do`

If `\robust@command@act@chk@args` branched to false, then `\robust@command@act@loop` will loop over the list of items in the ⟨*action-list*⟩ (`#1`), and process each item as described earlier. If the ⟨*if-type-n*⟩ command expands to ⟨*true*⟩ then `\robust@command@act@do` is used to execute ⟨*act-type-n*⟩ on the ⟨*act-arg*⟩, otherwise the loop resumes with the next item.

```
\long\def\robust@command@act@loop#1#2{\robust@command@act@loop@aux#1#2}
\long\def\robust@command@act@loop@aux#1#2#3{%
  \ifx\@nnil#2%
  \else
    #2{#1}%
      {\robust@command@act@do{#3}}%
      {\expandafter\robust@command@act@loop\expandafter#1}%
  \fi}
\long\def\robust@command@act@do#1%
  \fi#2%
  \robust@command@act@end#3#4{%
  \fi
  #1#4}
```

If the end is reached and no action was taken, then do ⟨*fallback-action*⟩⟨*act-arg*⟩.

`\robust@command@act@end`

```
\long\def\robust@command@act@end#1#2{#1#2}
```

`\robust@command@chk@safe`
`\robust@command@act@chk@args`

```
\long\def\robust@command@chk@safe#1{%
  \begingroup
    \escapechar=`\\
  \expandafter\endgroup\expandafter
  \robust@command@act@chk@args\meaning#1:->\@nil}
\def\robust@command@act@chk@args#1:->#2\@nil{%
  \@expl@str@if@eq@@nnTF{#1}{macro}%
    {\@firstoftwo}%
    {\@expl@str@if@eq@@nnTF{#1}{\protected macro}%
      {\@firstoftwo}%
      {\@secondoftwo}}}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\robust@command@act}
⟨latexrelease⟩  {Add \robust@command@act}%
⟨latexrelease⟩\let\robust@command@act\@undefined
⟨latexrelease⟩\let\robust@command@act@loop\@undefined
⟨latexrelease⟩\let\robust@command@act@loop@aux\@undefined
```

```
⟨latexrelease⟩\let\robust@command@act@do\@undefined
⟨latexrelease⟩\let\robust@command@act@end\@undefined
⟨latexrelease⟩\let\robust@command@chk@safe\@undefined
⟨latexrelease⟩\let\robust@command@act@chk@args\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\robust@command@act` *and others. These functions are documented on page* **??**.)

1.5.1 Copying robust commands

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020-10-01}{\DeclareCommandCopy}
⟨latexrelease⟩  {Add \NewCommandCopy, \RenewCommandCopy, and \DeclareCommandCopy}%
⟨*2ekernel | latexrelease⟩
```

`\NewCommandCopy`
`\RenewCommandCopy`
`\DeclareCommandCopy`

`\NewCommandCopy` starts by checking if `#1` is already defined, and raises an error if so, otherwise the definition is carried out. `\RenewCommandCopy` does (almost) the opposite. If the command is *not* defined, then an error is raised. But the definition is carried out anyhow, so the behaviour is consistent with `\renewcommand`.

A `\ProvideCommandCopy` isn't defined because it's not reasonably useful. `\provide...` commands mean "define this if there's no other definition", but copying a command (usually) implies that the command being copied is defined, so `\ProvideCommandCopy` doesn't make a lot of sense. But more importantly, the most common use case of copying a command is to redefine it later, while preserving the old definition, as in:

```
\ProvideComandCopy \A \B
\renewcommand \B { ... \A ... }
```

then, if `\A` is already defined the first line is skipped, an in this case `\B` won't work as expected.

The three versions call the internal `\declare@commandcopy` with the proper action. `\@firstofone` will carry out the copy. The only case when the copy is not made is the ⟨*false*⟩ case for `\NewCommandCopy`, in which the command already exists and the definition is aborted.

```
\def\NewCommandCopy{%
  \declare@commandcopy
    {\@firstofone}%
    {\@firstoftwo\@notdefinable}}
\def\RenewCommandCopy{%
  \declare@commandcopy
    {\@latex@error{Command \@backslashchar\reserved@a\space undefined}\@ehc
     \@firstofone}%
    {\@firstofone}}
\def\DeclareCommandCopy{%
  \declare@commandcopy
    {\@firstofone}%
    {\@firstofone}}
```

`\declare@commandcopy`

Start by checking if the command is already defined. The proper action is taken by each specific command above. If all's good, then `\robust@command@act` is called with the proper arguments as described earlier, with `\@declarecommandcopylisthook` as the ⟨*action-list*⟩ and `\declare@commandcopy@let` as the ⟨*fallback-action*⟩.

```
\long\def\declare@commandcopy#1#2#3#4{%
  \edef\reserved@a{\@expl@cs@to@str@@N#3}%
  \@ifundefined\reserved@a{#1}{#2}%
    {\robust@command@act
       \@declarecommandcopylisthook#4%
       \declare@commandcopy@let{#3#4}}}
```

The initial definition of `\@declarecommandcopylisthook` contains the tests for the two types of robust command in the kernel.

`\@declarecommandcopylisthook`

```
\def\@declarecommandcopylisthook{%
  {\@if@DeclareRobustCommand \@copy@DeclareRobustCommand}%
  {\@if@newcommand \@copy@newcommand}}
```

The initial definition of `\@declarecommandcopylisthook` contains the tests for the two types of robust command in the kernel.

`\declare@commandcopy@let`

```
\long\def\declare@commandcopy@let#1#2{\let#1=#2\relax}
```

Now the rollback code.

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\DeclareCommandCopy}
⟨latexrelease⟩  {Undefine \NewCommandCopy, \RenewCommandCopy, and \DeclareCommandCopy}%
⟨latexrelease⟩\let\NewCommandCopy\@undefined
⟨latexrelease⟩\let\RenewCommandCopy\@undefined
⟨latexrelease⟩\let\DeclareCommandCopy\@undefined
⟨latexrelease⟩\let\declare@commandcopy\@undefined
⟨latexrelease⟩\let\@declarecommandcopylisthook\@undefined
⟨latexrelease⟩\let\declare@commandcopy@let\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\NewCommandCopy` *and others. These functions are documented on page* **??**.)

1.5.2 Showing robust commands

`\ShowCommand`

Most of the machinery defined for `\NewCommandCopy` can be used to show the definition of a robust command, in a similar fashion to `texdef`. The difference is that after the command's is detected to has a given type of robustness, rather than making a copy, we use a separate routine to show its definition.

With all the machinery in place, `\ShowCommand` itself is quite simple: use `\robust@command@act` to iterate through the `\@showcommandlisthook` list, and if nothing is found, fallback to `\show`.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020-10-01}{\ShowCommand}%
⟨latexrelease⟩  {Add \ShowCommand}%
⟨*2ekernel | latexrelease⟩
```

```
\long\def\ShowCommand#1{%
  \robust@command@act
    \@showcommandlisthook#1%
    \show#1}
```

\@showcommandlisthook

The initial definition of \@showcommandlisthook contains the same tests as used for copying, but \@show@... commands instead of \@copy@.... Same as before, it is initialised to cope with \DeclareRobustCommand and \newcommand with optional arguments.

```
\def\@showcommandlisthook{%
  {\@if@DeclareRobustCommand \@show@DeclareRobustCommand}%
  {\@if@newcommand \@show@newcommand}}
```

Now the rollback code.

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\ShowCommand}
⟨latexrelease⟩  {Undefine \ShowCommand}%
⟨latexrelease⟩\let\ShowCommand\@undefined
⟨latexrelease⟩\let\@showcommandlisthook\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \ShowCommand *and* \@showcommandlisthook*. These functions are documented on page* ??.)

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020-10-01}{\@if@DeclareRobustCommand}
⟨latexrelease⟩  {Add \@if@DeclareRobustCommand, \@if@newcommand,
⟨latexrelease⟩       \@copy@DeclareRobustCommand, \@copy@newcommand,
⟨latexrelease⟩       \@show@DeclareRobustCommand, \@show@newcommand}%
⟨*2ekernel | latexrelease⟩
```

1.5.3 Commands defined with \DeclareRobustCommand

\@if@DeclareRobustCommand

Now that we provided a generic way to copy one macro to another, we need to define a way to check if a command is one of LaTeX 2_ε's robust types. These tests are heavily based on Heiko's \LetLtxMacro, but chopped into separate macros.

\@if@DeclareRobustCommand checks if a command \cmd was defined by \DeclareRobustCommand. The test returns true if the expansion of \cmd is exactly \protect\cmd␣.

```
\long\def\@if@DeclareRobustCommand#1{%
  \begingroup
    \escapechar=`\\
    \edef\reserved@a{\string#1}%
    \edef\reserved@b{\detokenize{#1}}%
    \xdef\@gtempa{%
      \ifx\reserved@a\reserved@b
         \noexpand\x@protect
         \noexpand#1%
      \fi
      \noexpand\protect
      \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname}%
  \endgroup
  \ifx\@gtempa#1\relax
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
```

\@copy@DeclareRobustCommand
\copy@kernel@robust@command

If a command was defined by \DeclareRobustCommand (that is, \@if@DeclareRobustCommand returns true), then to make a copy of \cmd into \foo we define the latter such that it expands to \protect\foo␣, then make \foo␣ equal to \cmd␣.

There is one detail we need to take care of: if a command was defined with \DeclareRobustCommand it may still have an optional argument, in which case there is one more macro layer before the actual definition of the command. We use \@if@newcommand to check that and \@copy@newcommand to do the copying.

```
\long\def\@copy@DeclareRobustCommand#1#2{%
  \begingroup
    \escapechar=`\\
    \edef\reserved@a{\string#1}%
    \edef\reserved@b{\detokenize{#1}}%
    \edef\reserved@a{%
  \endgroup
  \def\noexpand#1{%
    \ifx\reserved@a\reserved@b
       \noexpand\x@protect
       \noexpand#1%
    \fi
    \noexpand\protect
    \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname}%
  \noexpand\copy@kernel@robust@command
    \expandafter\noexpand\csname\@expl@cs@to@str@@N#1 \endcsname
    \expandafter\noexpand\csname\@expl@cs@to@str@@N#2 \endcsname}%
  \reserved@a}
\long\def\copy@kernel@robust@command#1#2{%
  \robust@command@chk@safe#2%
    {\@if@newcommand#2%
       {\@copy@newcommand}%
       {\declare@commandcopy@let}}
    {\declare@commandcopy@let}%
  #1#2}
```

\@show@DeclareRobustCommand
\show@kernel@robust@command

Showing the command is pretty simple. This command prints the top-level expansion as TEX's \show would, but with robust macro: rather than just macro:, then a blank line and then \show the inner command. For a macro defined with, say, \DeclareRobustCommand\foo[1]{bar}, it will print:

```
> \foo=robust macro:
->\protect \foo  .

> \foo =\long macro:
#1->bar.
```

If the inner command is defined with an optional argument, then \@show@newcommand is also used.

The value of \escapechar is deliberately not enforced, so \ShowCommand behaves more like \show.

```
\long\def\@show@DeclareRobustCommand#1{%
  \typeout{> \string#1=robust macro:}%
  \typeout{->\@expl@cs@replacement@spec@@N#1.^^J}%
  \expandafter\show@kernel@robust@command
    \csname\@expl@cs@to@str@@N#1 \endcsname}
```

```
\long\def\show@kernel@robust@command#1{%
  \robust@command@chk@safe#1%
    {\@if@newcommand#1%
       {\@show@newcommand}%
       {\show}}%
    {\show}%
  #1}
```

(*End definition for* `\@if@DeclareRobustCommand` *and others. These functions are documented on page* ??.)

1.5.4 Commands defined with `\newcommand` (with optional argument)

`\@if@newcommand` A command `\cmd` (or `\cmd␣`, if it was defined with `\DeclareRobustCommand`) with an optional argument will expand to `\@protected@testopt\cmd\\cmd{<opt>}`. To check that we look at the first three tokens in the expansion of `\cmd`, and return true or false accordingly.

This test *requires* that the command be a parameterless macro, otherwise it will not work (and probably break). This is ensured with `\robust@command@chk@safe` before calling `\@if@newcommand`.

```
\long\def\@if@newcommand#1{%
  \edef\reserved@a{%
    \noexpand\@protected@testopt
    \noexpand#1%
    \expandafter\noexpand\csname\@backslashchar\@expl@cs@to@str@@N#1\endcsname}%
  \edef\reserved@b{%
    \unexpanded\expandafter\expandafter\expandafter
      {\expandafter\@carcube#1{}{}{}\@nil}}%
  \ifx\reserved@a\reserved@b
    \expandafter\@firstoftwo
  \else
    \expandafter\@secondoftwo
  \fi}
```

`\@copy@newcommand` Then, if a command `\cmd` takes an optional argument, we copy it to `\foo` by defining the latter to expand to `\@protected@testopt\foo\\foo{<opt>}`.

```
\long\def\@copy@newcommand#1#2{%
  \edef#1{\noexpand\@protected@testopt
    \noexpand#1%
    \expandafter\noexpand\csname\@backslashchar\@expl@cs@to@str@@N#1\endcsname
    \unexpanded\expandafter\expandafter\expandafter
      {\expandafter\@gobblethree#2}}%
  \expandafter
  \let\csname\@backslashchar\@expl@cs@to@str@@N#1\expandafter\endcsname
      \csname\@backslashchar\@expl@cs@to@str@@N#2\endcsname}
```

`\@show@newcommand` `\@show@newcommand@aux` A command being `\shown` here is guaranteed to have an optional argument. Start by showing the top-level expansion of the command (using `\typeout` to avoid TeX asking for interaction and extra context lines), then call `\@show@newcommand@aux` with the internal command, which contains the actual definition, and with the expansion of the command to extract the default value of the optional argument.

```
\long\def\@show@newcommand#1{%
  \typeout{> \string#1=robust macro:}%
```

```
  \typeout{->\@expl@cs@replacement@spec@@N#1.^^J}%
  \expandafter\@show@newcommand@aux
    \csname\@backslashchar\@expl@cs@to@str@@N#1\expandafter\endcsname
    \expandafter{#1}}
```

For a macro defined with, say, `\newcommand\foo[1][opt]{bar}`, it will print:

```
> \foo=robust macro:
->\@protected@testopt \foo \\foo {opt}.

> \\foo=\long macro:
> default #1=opt.
[#1]->bar.
```

If the command was defined with `\DeclareRobustCommand`, then another pair of lines show the top-level expansion `\protect␣\foo␣␣`.

The extra gymnastics with `\showtokens` ensures that `\showtokens` itself, and the internals of this macro aren't showed in the context lines.

```
\long\def\@show@newcommand@aux#1#2{%
  \typeout{> \string#1=\@expl@cs@prefix@spec@@N#1macro:}%
  \edef\reserved@a{%
    default \string##1=\expandafter\detokenize\@gobblethree#2.^^J%
    \@expl@cs@argument@spec@@N#1->\@expl@cs@replacement@spec@@N#1}%
  \showtokens\expandafter\expandafter\expandafter{\expandafter\reserved@a}}
```

Now the rollback code.

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\@if@DeclareRobustCommand}
⟨latexrelease⟩  {Undefine \@if@DeclareRobustCommand, \@if@newcommand,
⟨latexrelease⟩            \@copy@DeclareRobustCommand, \@copy@newcommand,
⟨latexrelease⟩            \@show@DeclareRobustCommand, \@show@newcommand}%
⟨latexrelease⟩\let\@if@DeclareRobustCommand\@undefined
⟨latexrelease⟩\let\@copy@DeclareRobustCommand\@undefined
⟨latexrelease⟩\let\@show@DeclareRobustCommand\@undefined
⟨latexrelease⟩\let\@if@newcommand\@undefined
⟨latexrelease⟩\let\@copy@newcommand\@undefined
⟨latexrelease⟩\let\@show@newcommand\@undefined
%
⟨latexrelease⟩\let\copy@kernel@robust@command\@undefined
⟨latexrelease⟩\let\show@kernel@robust@command\@undefined
⟨latexrelease⟩\let\@show@newcommand@aux\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@if@newcommand` *and others. These functions are documented on page* **??**.)

1.6 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,

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2018-04-01}{\@ifundefined}
```

```
⟨latexrelease⟩{Leave commands undefined in \@ifundefined}%
⟨*2ekernel | latexrelease⟩
```

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

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

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

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

Now test of engine.

```
\ifx\numexpr\@undefined
```

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

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

Use the `\ifcsname` defined above.

```
\else
```

Optimised version for LuaTEX, using `\lastnamedcs`

```
\def\@ifundefined#1{%
  \ifcsname#1\endcsname
    \expandafter\ifx\lastnamedcs\relax\else\@ifundefin@d@i\fi
  \fi
  \@firstoftwo}

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

\fi
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\@ifundefined}
⟨latexrelease⟩{Leave commands undefined in \@ifundefined}%
⟨latexrelease⟩\def\@ifundefined#1{%
⟨latexrelease⟩  \expandafter\ifx\csname#1\endcsname\relax
⟨latexrelease⟩    \expandafter\@firstoftwo
⟨latexrelease⟩  \else
⟨latexrelease⟩    \expandafter\@secondoftwo
⟨latexrelease⟩  \fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@ifundefined`*. This function is documented on page* **??**.)

\@qend
\@qrelax

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

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

(*End definition for* \@qend *and* \@qrelax*. These functions are documented on page* ??.)

\@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}
```

(*End definition for* \@ifnextchar*. This function is documented on page* ??.)

\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
```

(*End definition for* \kernel@ifnextchar*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@ifnch*. This function is documented on page* ??.)

\@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
```

(*End definition for* \@sptoken*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@xifnch*. This function is documented on page* ??.)

\@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}}}
```

(*End definition for* \@ifstar*. This function is documented on page* ??.)

\@dblarg
\@xdblarg

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

(*End definition for* \@dblarg *and* \@xdblarg*. These functions are documented on page* ??.)

\@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\~}
```

(*End definition for* \@sanitize*. This function is documented on page* ??.)

\@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}%
}
```

(*End definition for* \@onelevel@sanitize*. This function is documented on page* ??.)

\string@makeletter
\@string@makeletter
\char@if@alph Iterates through a string, turning each alphabetic character into a catcode-11 token (partly undoes a \detokenize). Useful for \ifx-based string comparisons where \detokenize-ing the other string would break too much code.

The macro uses expl3's \@expl@str@map@function@@NN to iterate on the string (without losing spaces) and applies \@string@makeletter on each character. The latter checks if character is between a–z or A–Z, and uses \@alph or \@Alph to get the corresponding catcode-11 token. Other tokens are passed through unchanged.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\string@makeletter}
⟨latexrelease⟩  {Add \string@makeletter}%
⟨*2ekernel | latexrelease⟩
\def\string@makeletter#1{%
  \@expl@str@map@function@@NN#1\@string@makeletter}
\def\@string@makeletter#1{%
```

```
  \char@if@alph{#1}%
    {\@expl@char@generate@@nn{`#1}{11}}%
    {#1}}
\def\char@if@alph#1{%
  \ifnum0\ifnum`#1<`A 1\fi\ifnum`#1>`z 1\fi
      \if\ifnum`#1>`Z @\fi\ifnum`#1<`a @\fi01\fi>0
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\string@makeletter}
⟨latexrelease⟩  {Undefine \string@makeletter}%
⟨latexrelease⟩\let\string@makeletter\@undefined
⟨latexrelease⟩\let\@string@makeletter\@undefined
⟨latexrelease⟩\let\char@if@alph\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\string@makeletter`, `\@string@makeletter`, *and* `\char@if@alph`. *These functions are documented on page* **??**.)

`\makeatletter` `\makeatother`

Make internal control sequences accessible or inaccessible.

```
\DeclareRobustCommand\makeatletter{\catcode`\@11\relax}
\DeclareRobustCommand\makeatother{\catcode`\@12\relax}
```

(*End definition for* `\makeatletter` *and* `\makeatother`. *These functions are documented on page* **??**.)

2 Discretionary Hyphenation

`\-` `\@dischyph`

Moved here to be after the definition of `\DeclareRobustCommand`.

The primitive `\-` command adds a discretionary hyphen using the current font's `\hyphenchar`. Monospace fonts are usually declared with `\hyphenchar` set to −1 to suppress hyhenation.

LaTeX, from LaTeX2.09 in 1986 defined `\-` by

```
\def\-{\discretionary{-}{}{}}
```

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.

LaTeX 2_ε, between 1993 and 2017, had a comment at this point saying that the definition "would probably change" because the definition always uses `-`. The definition used below was given in comments at this point during time.

In 2017 we finally enabled this definition by default, with the older LaTeX definition accessible via latexrelease as usual.

In LuaLaTeX the primitive definition of `\-` is used directly because it's use of extended hyphenation parameters means that `\-` works correctly even with `\hyphenchar` set to -1. This change makes `\-` under LuaLaTeX compatible with language specific hyphenation characters.

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

```
\def\@latex@info#1{}
```

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\-}{Use primitive \- in Lua\LaTeX}%
⟨*2ekernel | latexrelease⟩
\ifx\directlua\@undefined
  \DeclareRobustCommand{\-}{%
    \discretionary{%
      \char \ifnum\hyphenchar\font<\z@
              \defaulthyphenchar
            \else
              \hyphenchar\font
            \fi
                }{}{}%
  }
\else
  \let\-\@@hyph
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2017/04/15}{\-}{Use \hyphenchar in \-}%
⟨latexrelease⟩\DeclareRobustCommand{\-}{%
⟨latexrelease⟩  \discretionary{%
⟨latexrelease⟩    \char \ifnum\hyphenchar\font<\z@
⟨latexrelease⟩            \defaulthyphenchar
⟨latexrelease⟩          \else
⟨latexrelease⟩            \hyphenchar\font
⟨latexrelease⟩          \fi
⟨latexrelease⟩               }{}{}%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\-}{Use \hyphenchar in \-}%
⟨latexrelease⟩\def\-{\discretionary{-}{}{}}
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨*2ekernel | latexrelease⟩
\let\@dischyph=\-
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
```

(*End definition for* `\-` *and* `\@dischyph`*. These functions are documented on page* **??**.)

Delayed from `ltvers.dtx`

```
\newif\if@includeinrelease
\@includeinreleasefalse
```

Delayed from `ltplain.dtx`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\allowbreak}{Make various commands robust}%
```

```
\MakeRobust\allowbreak
\MakeRobust\bigbreak
\MakeRobust\break
\MakeRobust\dotfill
\MakeRobust\frenchspacing
\MakeRobust\goodbreak
\MakeRobust\hrulefill
\MakeRobust\medbreak
\MakeRobust\nobreak
\MakeRobust\nonfrenchspacing
\MakeRobust\obeylines
\MakeRobust\obeyspaces
\MakeRobust\slash
\MakeRobust\smallbreak
\MakeRobust\strut
\MakeRobust\underbar
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\allowbreak}{Make various commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\allowbreak
⟨latexrelease⟩\kernel@make@fragile\bigbreak
⟨latexrelease⟩\kernel@make@fragile\break
⟨latexrelease⟩\kernel@make@fragile\dotfill
⟨latexrelease⟩\kernel@make@fragile\frenchspacing
⟨latexrelease⟩\kernel@make@fragile\goodbreak
⟨latexrelease⟩\kernel@make@fragile\hrulefill
⟨latexrelease⟩\kernel@make@fragile\medbreak
⟨latexrelease⟩\kernel@make@fragile\nobreak
⟨latexrelease⟩\kernel@make@fragile\nonfrenchspacing
⟨latexrelease⟩\kernel@make@fragile\obeylines
⟨latexrelease⟩\kernel@make@fragile\obeyspaces
⟨latexrelease⟩\kernel@make@fragile\slash
⟨latexrelease⟩\kernel@make@fragile\smallbreak
⟨latexrelease⟩\kernel@make@fragile\strut
⟨latexrelease⟩\kernel@make@fragile\underbar
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\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}
```

(*End definition for* `\g@addto@macro`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
```

File g
lthooks.dtx

Contents

1 Introduction

Hooks are points in the code of commands or environments where it is possible to add processing code into existing commands. This can be done by different packages that do not know about each other and to allow for hopefully safe processing it is necessary to sort different chunks of code added by different packages into a suitable processing order.

This is done by the packages adding chunks of code (via `\AddToHook`) and labeling their code with some label by default using the package name as a label.

At `\begin{document}` all code for a hook is then sorted according to some rules (given by `\DeclareHookRule`) for fast execution without processing overhead. If the hook code is modified afterwards (or the rules are changed), a new version for fast processing is generated.

Some hooks are used already in the preamble of the document. If that happens then the hook is prepared for execution (and sorted) already at that point.

2 Package writer interface

The hook management system is offered as a set of CamelCase commands for traditional LaTeX 2_ε packages (and for use in the document preamble if needed) as well as `expl3` commands for modern packages, that use the L3 programming layer of LaTeX. Behind the scenes, a single set of data structures is accessed so that packages from both worlds can coexist and access hooks in other packages.

2.1 LaTeX 2_ε interfaces

2.1.1 Declaring hooks and using them in code

With two exceptions, hooks have to be declared before they can be used. The exceptions are hooks in environments (i.e., executed at `\begin` and `\end`) and hooks run when loading files, e.g. before and after a package is loaded, etc. Their hook names depend on the environment or the file name and so declaring them beforehand is difficult.

`\NewHook` `\NewHook {⟨hook⟩}`

Creates a new ⟨*hook*⟩. If this is a hook provided as part of a package it is suggested that the ⟨*hook*⟩ name is always structured as follows: ⟨*package-name*⟩`/`⟨*hook-name*⟩. If necessary you can further subdivide the name by adding more `/` parts. If a hook name is already taken, an error is raised and the hook is not created.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

`\NewReversedHook` `\NewReversedHook {⟨hook⟩}`

Like `\NewHook` declares a new ⟨*hook*⟩. the difference is that the code chunks for this hook are in reverse order by default (those added last are executed first). Any rules for the hook are applied after the default ordering. See sections 2.3 and 2.4 for further details.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

`\NewMirroredHookPair` `\NewMirroredHookPair {⟨hook-1⟩} {⟨hook-2⟩}`

A shorthand for `\NewHook{⟨hook-1⟩}\NewReversedHook{⟨hook-2⟩}`.

The ⟨*hooks*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

`\UseHook` `\UseHook {⟨hook⟩}`

Execute the hook code inside a command or environment.

Before `\begin{document}` the fast execution code for a hook is not set up, so in order to use a hook there it is explicitly initialized first. As that involves assignments using a hook at those times is not 100% the same as using it after `\begin{document}`.

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading `.` is treated literally.

`\UseOneTimeHook` `\UseOneTimeHook {⟨hook⟩}`

Some hooks are only used (and can be only used) in one place, for example, those in `\begin{document}` or `\end{document}`. Once we have passed that point adding to the hook through a defined `\⟨addto-cmd⟩` command (e.g., `\AddToHook` or `\AtBeginDocument`, etc.) would have no effect (as would the use of such a command inside the hook code itself). It is therefore customary to redefine `\⟨addto-cmd⟩` to simply process its argument, i.e., essentially make it behave like `\@firstofone`.

`\UseOneTimeHook` does that: it records that the hook has been consumed and any further attempt to add to it will result in executing the code to be added immediately.

FMi: Maybe add an error version as well?

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading `.` is treated literally.

2.1.2 Updating code for hooks

\AddToHook

\AddToHook {⟨*hook*⟩}[⟨*label*⟩]{⟨*code*⟩}

Adds ⟨*code*⟩ to the ⟨*hook*⟩ labeled by ⟨*label*⟩. If the optional argument ⟨*label*⟩ is not provided, if \AddToHook is used in a package/class, then the current package/class name is used, otherwise top-level is used (see section 2.1.3).

If there already exists code under the ⟨*label*⟩ then the new ⟨*code*⟩ is appended to the existing one (even if this is a reversed hook). If you want to replace existing code under the ⟨*label*⟩, first apply \RemoveFromHook.

The hook doesn't have to exist for code to be added to it. However, if it is not declared later then obviously the added ⟨*code*⟩ will never be executed. This allows for hooks to work regardless of package loading order and enables packages to add to hook of other packages without worrying whether they are actually used in the current document. See section 2.1.5.

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\RemoveFromHook

\RemoveFromHook {⟨*hook*⟩}[⟨*label*⟩]

Removes any code labeled by ⟨*label*⟩ from the ⟨*hook*⟩. If the optional argument ⟨*label*⟩ is not provided, if \AddToHook is used in a package/class, then the current package/class name is used, otherwise top-level is used.

If the code for that ⟨*label*⟩ wasn't yet added to the ⟨*hook*⟩, an order is set so that when some code attempts to add that label, the removal order takes action and the code is not added.

If the optional argument is *, then all code chunks are removed. This is rather dangerous as it drops code from other packages one may not know about!

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

In contrast to the voids relationship between two labels in a \DeclareHookrule this is a destructive operation as the labeled code is removed from the hook data structure, whereas the relationship setting can be undone by providing a different relationship later.

A useful application for this declaration inside the document body is when one wants to temporarily add code to hooks and later remove it again, e.g.,

```
\AddToHook{env/quote/before}{\small}
\begin{quote}
  A quote set in a smaller typeface
\end{quote}
...
\RemoveFromHook{env/quote/before}
... now back to normal for further quotes
```

Note that you can't cancel the setting with

```
\AddToHook{env/quote/before}{}
```

because that only "adds" a further empty chunk of code to the hook. Adding \normalsize would work but that means the hook then contained \small\normalsize which means to font size changes for no good reason.

The above is only needed if one wants to typeset several quotes in a smaller typeface. If the hook is only needed once then `\AddToHookNext` is simpler, because it resets itself after one use.

`\AddToHookNext` `\AddToHookNext {⟨hook⟩}{⟨code⟩}`

Adds ⟨*code*⟩ to the next invocation of the ⟨*hook*⟩. The code is executed after the normal hook code has finished and it is executed only once, i.e. it is deleted after it was used.

Using the declaration is a global operation, i.e., the code is not lost, even if the declaration is used inside a group and the next invocation happens after the group. If the declaration is used several times before the hook is executed then all code is executed in the order in which it was declared.[2]

It is possible to nest declarations using the same hook (or different hooks), e.g.,

`\AddToHookNext{⟨hook⟩}{⟨code-1⟩\AddToHookNext{⟨hook⟩}{⟨code-2⟩}}`

will execute ⟨*code-1*⟩ next time the ⟨*hook*⟩ is used and at that point puts ⟨*code-2*⟩ into the ⟨*hook*⟩ so that it gets executed on following time the hook is run.

A hook doesn't have to exist for code to be added to it. This allows for hooks to work regardless of package loading order. See section 2.1.5.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

2.1.3 Hook names and default labels

It is best practice to use `\AddToHook` in packages or classes *without specifying a* ⟨*label*⟩ because then the package or class name is automatically used, which is helpful if rules are needed, and avoids mistyping the ⟨*label*⟩.

Using an explicit ⟨*label*⟩ is only necessary in very specific situations, e.g., if you want to add several chunks of code into a single hook and have them placed in different parts of the hook (by providing some rules).

The other case is when you develop a larger package with several sub-packages. In that case you may want to use the same ⟨*label*⟩ throughout the sub-packages in order to avoid that the labels change if you internally reorganize your code.

Except for `\UseHook`, `\UseOneTimeHook`, `\IfHookEmptyTF`, and `\IfHookExistsTF` (and their expl3 interfaces `\hook_use:n`, `\hook_use_once:n`, `\hook_if_empty:nTF`, and `\hook_if_exist:nTF`), all ⟨*hook*⟩ and ⟨*label*⟩ arguments are processed in the same way: first, spaces are trimmed around the argument, then it is fully expanded until only character tokens remain. If the full expansion of the ⟨*hook*⟩ or ⟨*label*⟩ contains a non-expandable non-character token, a low-level TEX error is raised (namely, the ⟨*hook*⟩ is expanded using TEX's `\csname`...`\endcsname`, as such, Unicode characters are allowed in ⟨*hook*⟩ and ⟨*label*⟩ arguments). The arguments of `\UseHook` and `\UseOneTimeHook` are processed much in the same way except that spaces are not trimmed around the argument, for better performance.

It is not enforced, but highly recommended that the hooks defined by a package, and the ⟨*labels*⟩ used to add code to other hooks contain the package name to easily identify the source of the code chunk and to prevent clashes. This should be the standard practice, so this hook management code provides a shortcut to refer to the current package in the name of a ⟨*hook*⟩ and in a ⟨*label*⟩. If ⟨*hook*⟩ name or ⟨*label*⟩ consist just of a single dot (`.`), or starts with a dot followed by a slash (`./`) then the dot denotes the ⟨*default label*⟩

[2] There is no mechanism to reorder such code chunks (or delete them).

(usually the current package or class name—see `\DeclareDefaultHookLabel`). A "`.`" or "`./`" anywhere else in a ⟨*hook*⟩ or in ⟨*label*⟩ is treated literally and is not replaced.

For example, inside the package `mypackage.sty`, the default label is `mypackage`, so the instructions:

```
\NewHook   {./hook}
\AddToHook {./hook}[.]{code}     % Same as \AddToHook{./hook}{code}
\AddToHook {./hook}[./sub]{code}
\DeclareHookRule{begindocument}{.}{before}{babel}
\AddToHook {file/after/foo.tex}{code}
```

are equivalent to:

```
\NewHook   {mypackage/hook}
\AddToHook {mypackage/hook}[mypackage]{code}
\AddToHook {mypackage/hook}[mypackage/sub]{code}
\DeclareHookRule{begindocument}{mypackage}{before}{babel}
\AddToHook {file/after/foo.tex}{code}                  % unchanged
```

The ⟨*default label*⟩ is automatically set to the name of the current package or class (using `\@currname`). If `\@currname` is not set (because the hook command is used outside of a package, or the current file wasn't loaded with `\usepackage` or `\documentclass`), then the `top-level` is used as the ⟨*default label*⟩.

This syntax is available in all ⟨*label*⟩ arguments and most ⟨*hook*⟩, both in the LaTeX 2_ε interface, and the LaTeX3 interface described in section 2.2.

Note, however, that the replacement of `.` by the ⟨*default label*⟩ takes place when the hook command is executed, so actions that are somehow executed after the package ends will have the wrong ⟨*default label*⟩ if the dot-syntax is used. For that reason, this syntax is not available in `\UseHook` (and `\hook_use:n`) because the hook is most of the time used outside of the package file in which it was defined. This syntax is also not available in the hook conditionals `\IfHookEmptyTF` (and `\hook_if_empty:nTF`) and `\IfHookExistsTF` (and `\hook_if_exist:nTF`) because these conditionals are used in some performance-critical parts of the hook management code, and because they are usually used to refer to other package's hooks, so the dot-syntax doesn't make much sense.

In some cases, for example in large packages, one may want to separate it in logical parts, but still use the main package name as ⟨*label*⟩, then the ⟨*default label*⟩ can be set using `\DeclareDefaultHookLabel`:

`\DeclareDefaultHookLabel`

`\DeclareDefaultHookLabel {`⟨*default label*⟩`}`

Sets the ⟨*default label*⟩ to be used in ⟨*label*⟩ arguments. If `\DeclareDefaultHookLabel` is not used in the current package, `\@currname` is used instead. If `\@currname` is not set, the code is assumed to be in the main document, in which case `top-level` is used.

The effect of `\DeclareDefaultHookLabel` holds for the current file, and is reset to the previous value when the file is closed.

2.1.4 Defining relations between hook code

The default assumption is that code added to hooks by different packages is independent and the order in which it is executed is irrelevant. While this is true in many case it is obviously false in many others.

Before the hook management system was introduced packages had to take elaborate precaution to determine of some other package got loaded as well (before or after) and find some ways to alter its behavior accordingly. In addition is was often the user's responsibility to load packages in the right order so that code added to hooks got added in the right order and some cases even altering the loading order wouldn't resolve the conflicts.

With the new hook management system it is now possible to define rules (i.e., relationships) between code chunks added by different packages and explicitly describe in which order they should be processed.

`\DeclareHookRule` `\DeclareHookRule {⟨hook⟩}{⟨label1⟩}{⟨relation⟩}{⟨label2⟩}`

Defines a relation between ⟨*label1*⟩ and ⟨*label2*⟩ for a given ⟨*hook*⟩. If ⟨*hook*⟩ is `??` this defines a default relation for all hooks that use the two labels, i.e., that have chunks of code labeled with ⟨*label1*⟩ and ⟨*label2*⟩. Rules specific to a given hook take precedence over default rules that use `??` as the ⟨*hook*⟩.

Currently, the supported relations are the following:

`before` or `<` Code for ⟨*label1*⟩ comes before code for ⟨*label2*⟩.

`after` or `>` Code for ⟨*label1*⟩ comes after code for ⟨*label2*⟩.

`incompatible-warning` Only code for either ⟨*label1*⟩ or ⟨*label2*⟩ can appear for that hook (a way to say that two packages—or parts of them—are incompatible). A warning is raised if both labels appear in the same hook.

`incompatible-error` Like `incompatible-error` but instead of a warning a LaTeX error is raised, and the code for both labels are dropped from that hook until the conflict is resolved.

`voids` Code for ⟨*label1*⟩ overwrites code for ⟨*label2*⟩. More precisely, code for ⟨*label2*⟩ is dropped for that hook. This can be used, for example if one package is a superset in functionality of another one and therefore wants to undo code in some hook and replace it with its own version.

`unrelated` The order of code for ⟨*label1*⟩ and ⟨*label2*⟩ is irrelevant. This rule is there to undo an incorrect rule specified earlier.

There can only be a single relation between two labels for a given hook, i.e., a later `\DeclareHookrule` overwrites any previous delcaration.

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

`\ClearHookRule` `\ClearHookRule{⟨hook⟩}{⟨label1⟩}{⟨label2⟩}`

Syntactic sugar for saying that ⟨*label1*⟩ and ⟨*label2*⟩ are unrelated for the given ⟨*hook*⟩.

\DeclareDefaultHookRule

\DeclareDefaultHookRule{⟨*label1*⟩}{⟨*relation*⟩}{⟨*label2*⟩}

This sets up a relation between ⟨*label1*⟩ and ⟨*label2*⟩ for all hooks unless overwritten by a specific rule for a hook. Useful for cases where one package has a specific relation to some other package, e.g., is `incompatible` or always needs a special ordering `before` or `after`. (Technically it is just a shorthand for using `\DeclareHookRule` with `??` as the hook name.)

Declaring default rules is only supported in the document preamble.[3]

The ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

2.1.5 Querying hooks

Simpler data types, like token lists, have three possible states; they can:

- exist and be empty;
- exist and be non-empty; and
- not exist (in which case emptiness doesn't apply);

Hooks are a bit more complicated: they have four possible states. A hook may exist or not, and either way it may or may not be empty. This means that even a hook that doesn't exist may be non-empty.

This seemingly strange state may happen when, for example, package *A* defines hook `A/foo`, and package *B* adds some code to that hook. However, a document may load package *B* before package *A*, or may not load package *A* at all. In both cases some code is added to hook `A/foo` without that hook being defined yet, thus that hook is said to be non-empty, whereas it doesn't exist. Therefore, querying the existence of a hook doesn't imply its emptiness, neither does the other way around.

A hook is said to be empty when no code was added to it, either to its permanent code pool, or to its "next" token list. The hook doesn't need to be declared to have code added to its code pool. A hook is said to exist when it was declared with `\NewHook` or some variant thereof.

\IfHookEmptyTF ⋆

\IfHookEmptyTF {⟨*hook*⟩} {⟨*true code*⟩} {⟨*false code*⟩}

Tests if the ⟨*hook*⟩ is empty (*i.e.*, no code was added to it using either `\AddToHook` or `\AddToHookNext`), and branches to either ⟨*true code*⟩ or ⟨*false code*⟩ depending on the result.

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading `.` is treated literally.

[3] Trying to do so, e.g., via `\DeclareHookRule` with `??` has bad side-effects and is not supported (though not explicitly caught for performance reasons).

`\IfHookExistsTF` ⋆ `\IfHookExistsTF {⟨hook⟩} {⟨true code⟩} {⟨false code⟩}`

Tests if the ⟨*hook*⟩ exists (if it was created with either `\NewHook`, `\NewReversedHook`, or `\NewMirroredHookPair`), and branches to either ⟨*true code*⟩ or ⟨*false code*⟩ depending on the result.

The existence of a hook usually doesn't mean much from the viewpoint of code that tries to add/remove code from that hook, since package loading order may vary, thus the creation of hooks is asynchronous to adding and removing code from it, so this test should be used sparingly.

Generic hooks are declared at the time code is added to them, so the result of `\hook_if_exist:n` will change once code is added to said hook (unless the hook was previously declared).

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading `.` is treated literally.

FMi: Would be helpful if we provide some use cases

2.1.6 Displaying hook code

If one has to adjust the code execution in a hook using a hook rule it is helpful to get some information about the code associated with a hook, its current order and the existing rules.

\ShowHook
\LogHook

\ShowHook {⟨*hook*⟩}

Displays information about the ⟨*hook*⟩ such as

- the code chunks (and their labels) added to it,
- any rules set up to order them,
- the computed order in which the chunks are executed,
- any code executed on the next invocation only.

\LogHook prints the information to the .log file, and \ShowHook prints them to the terminal/command window and starts TEX's prompt (only in \errorstopmode) to wait for user action.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

Suppose a hook example-hook whose output of \ShowHook{example-hook} is:

```
1  -> The hook 'example-hook':
2  > Code chunks:
3  >     foo -> [code from package 'foo']
4  >     bar -> [from package 'bar']
5  >     baz -> [package 'baz' is here]
6  > Extra code for next invocation:
7  >     -> [one-time code]
8  > Rules:
9  >     foo|baz with relation >
10 >     baz|bar with default relation <
11 > Execution order (after applying rules):
12 >     bar, baz, foo.
```

In the listing above, lines 3 to 5 show the three code chunks added to the hook and their respective labels in the format

⟨*label*⟩ -> ⟨*code*⟩

Line 7 shows the code chunk for the next execution of the hook in the format

-> ⟨*next-code*⟩

This code will be used and disappear at the next \UseHook{example-hook}, in contrast to the chunks mentioned earlier, which can only be removed from that hook by doing \RemoveFromHook{⟨*label*⟩}[example-hook].

Lines 9 and 10 show the rules declared that affect this hook in the format

⟨*label-1*⟩|⟨*label-2*⟩ with ⟨*default?*⟩ relation ⟨*relation*⟩

which means that the ⟨*relation*⟩ applies to ⟨*label-1*⟩ and ⟨*label-2*⟩, in that order, as detailed in \DeclareHookRule. If the relation is default it means that that rule applies to ⟨*label-1*⟩ and ⟨*label-2*⟩ in *all* hooks, (unless overrided by a non-default relation).

Finally, line 12 lists the labels in the hook after sorting; that is, in the order they will be executed when the hook is used.

2.1.7 Debugging hook code

\DebugHooksOn
\DebugHooksOff

\DebugHooksOn

Turn the debugging of hook code on or off. This displays changes made to the hook data structures. The output is rather coarse and not really intended for normal use.

2.2 L3 programming layer (expl3) interfaces

This is a quick summary of the LaTeX3 programming interfaces for use with packages written in expl3. In contrast to the LaTeX 2ε interfaces they always use mandatory arguments only, e.g., you always have to specify the ⟨*label*⟩ for a code chunk. We therefore suggest to use the declarations discussed in the previous section even in expl3 packages, but the choice is yours.

\hook_new:n
\hook_new_reversed:n
\hook_new_pair:nn

\hook_new:n{⟨*hook*⟩}
\hook_new_pair:nn{⟨*hook-1*⟩}{⟨*hook-2*⟩}

Creates a new ⟨*hook*⟩ with normal or reverse ordering of code chunks. \hook_new_pair:nn creates a pair of such hooks with {⟨*hook-2*⟩} being a reversed hook. If a hook name is already taken, an error is raised and the hook is not created.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\hook_use:n

\hook_use:n {⟨*hook*⟩}

Executes the {⟨*hook*⟩} code followed (if set up) by the code for next invocation only, then empties that next invocation code.

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading . is treated literally.

\hook_use_once:n

\hook_use_once:n {⟨*hook*⟩}

Changes the {⟨*hook*⟩} status so that from now on any addition to the hook code is executed immediately. Then execute any {⟨*hook*⟩} code already set up.

FMi: better L3 name?

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading . is treated literally.

\hook_gput_code:nnn

\hook_gput_code:nnn {⟨*hook*⟩} {⟨*label*⟩} {⟨*code*⟩}

Adds a chunk of ⟨*code*⟩ to the ⟨*hook*⟩ labeled ⟨*label*⟩. If the label already exists the ⟨*code*⟩ is appended to the already existing code.

If code is added to an external ⟨*hook*⟩ (of the kernel or another package) then the convention is to use the package name as the ⟨*label*⟩ not some internal module name or some other arbitrary string.

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\hook_gput_next_code:nn

\hook_gput_next_code:nn {⟨hook⟩} {⟨code⟩}

Adds a chunk of ⟨*code*⟩ for use only in the next invocation of the ⟨*hook*⟩. Once used it is gone.

This is simpler than \hook_gput_code:nnn, the code is simply appended to the hook in the order of declaration at the very end, i.e., after all standard code for the hook got executed.

Thus if one needs to undo what the standard does one has to do that as part of ⟨*code*⟩.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\hook_gremove_code:nn

\hook_gremove_code:nn {⟨hook⟩} {⟨label⟩}

Removes any code for ⟨*hook*⟩ labeled ⟨*label*⟩.

If the code for that ⟨*label*⟩ wasn't yet added to the ⟨*hook*⟩, an order is set so that when some code attempts to add that label, the removal order takes action and the code is not added.

If the second argument is *, then all code chunks are removed. This is rather dangerous as it drops code from other packages one may not know about, so think twice before using that!

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\hook_gset_rule:nnnn

\hook_gset_rule:nnnn {⟨hook⟩} {⟨label1⟩} {⟨relation⟩} {⟨label2⟩}

Relate ⟨*label1*⟩ with ⟨*label2*⟩ when used in ⟨*hook*⟩. See \DeclareHookRule for the allowed ⟨*relation*⟩s. If ⟨*hook*⟩ is ?? a default rule is specified.

The ⟨*hook*⟩ and ⟨*label*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3. The dot-syntax is parsed in both ⟨*label*⟩ arguments, but it usually makes sense to be used in only one of them.

\hook_if_empty_p:n ⋆
\hook_if_empty:n*TF* ⋆

\hook_if_empty:nTF {⟨hook⟩} {⟨true code⟩} {⟨false code⟩}

Tests if the ⟨*hook*⟩ is empty (*i.e.*, no code was added to it using either \AddToHook or \AddToHookNext), and branches to either ⟨*true code*⟩ or ⟨*false code*⟩ depending on the result.

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading . is treated literally.

\hook_if_exist_p:n ⋆
\hook_if_exist:n*TF* ⋆

\hook_if_exist:nTF {⟨hook⟩} {⟨true code⟩} {⟨false code⟩}

Tests if the ⟨*hook*⟩ exists (if it was created with either \NewHook, \NewReversedHook, or \NewMirroredHookPair), and branches to either ⟨*true code*⟩ or ⟨*false code*⟩ depending on the result.

The existence of a hook usually doesn't mean much from the viewpoint of code that tries to add/remove code from that hook, since package loading order may vary, thus the creation of hooks is asynchronous to adding and removing code from it, so this test should be used sparingly.

Generic hooks are declared at the time code is added to them, so the result of \hook_if_exist:n will change once code is added to said hook (unless the hook was previously declared).

The ⟨*hook*⟩ *cannot* be specified using the dot-syntax. A leading . is treated literally.

\hook_show:n
\hook_log:n

\hook_show:n {⟨*hook*⟩}

Displays information about the ⟨*hook*⟩ such as

- the code chunks (and their labels) added to it,
- any rules set up to order them,
- the computed order in which the chunks are executed,
- any code executed on the next invocation only.

\hook_log:n prints the information to the .log file, and \hook_show:n prints them to the terminal/command window and starts TEX's prompt (only if \errorstopmode) to wait for user action.

The ⟨*hook*⟩ can be specified using the dot-syntax to denote the current package name. See section 2.1.3.

\hook_debug_on:
\hook_debug_off:

\hook_debug_on:

Turns the debugging of hook code on or off. This displays changes to the hook data.

2.3 On the order of hook code execution

Chunks of code for a ⟨*hook*⟩ under different labels are supposed to be independent if there are no special rules set up that define a relation between the chunks. This means that you can't make assumptions about the order of execution!

Suppose you have the following declarations:

```
\NewHook{myhook}
\AddToHook{myhook}[packageA]{\typeout{A}}
\AddToHook{myhook}[packageB]{\typeout{B}}
\AddToHook{myhook}[packageC]{\typeout{C}}
```

then executing the hook with \UseHook will produce the typeout A B C in that order. In other words, the execution order is computed to be packageA, packageB, packageC which you can verify with \ShowHook{myhook}:

```
The hook 'myhook':
 Code chunks:
    packageA -> \typeout {A}
    packageB -> \typeout {B}
    packageC -> \typeout {C}
 Extra code next invocation:
    ---
 Rules:
    ---
 Execution order:
    packageA, packageB, packageC
```

The reason is that the code chunks are internally saved in a property list and the initial order of such a property list is the order in which key-value pairs got added. However, that is only true if nothing other than adding happens!

Suppose, or example, you want to replace the code chunk for packageA, e.g.,

```
\RemoveFromHook{myhook}[packageA]
\AddToHook{myhook}[packageA]{\typeout{A alt}}
```

then your order becomes `packageB`, `packageC`, `packageA` because the label got removed from the property list and then re-added (at its end).

While that may not be too surprising, the execution order is also sometimes altered if you add a redundant rule, e.g. if you specify

```
\DeclareHookRule{myhook}{packageA}{before}{packageB}
```

instead of the previous lines we get

```
The hook 'myhook':
 Code chunks:
    packageA -> \typeout {A}
    packageB -> \typeout {B}
    packageC -> \typeout {C}
 Extra code next invocation:
    ---
 Rules:
    packageA|packageB with relation before
 Execution order (after applying rules):
    packageA, packageC, packageB
```

As you can see the code chunks are still in the same order, but in the execution order for the labels `packageB` and `packageC` have swapped places. The reason is that, with the rule there are two orders that satisfy it, and the algorithm for sorting happened to pick a different one compared to the case without rules (where it doesn't run at all as there is nothing to resolve). Incidentally, if we had instead specified the redundant rule

```
\DeclareHookRule{myhook}{packageB}{before}{label-3}
```

the execution order would not have changed.

In summary: it is not possible to rely on the order of execution unless there are rules that partially or fully define the order (in which you can rely on them being fulfilled).

2.4 The use of "reversed" hooks

You may have wondered why you can declare a "reversed" hook with `\NewReversedHook` and what that does exactly.

In short: the execution order of a reversed hook (without any rules!) is exactly reversed to the order you would have gotten for a hook declared with `\NewHook`.

This is helpful if you have a pair of hooks where you expect to see code added that involves grouping, e.g., starting an environment in the first and closing that environment in the second hook. To give a somewhat contrived example[4], suppose there is a package adding the following:

```
\AddToHook{env/quote/before}[package-1]{\begin{itshape}}
\AddToHook{env/quote/after} [package-1]{\end{itshape}}
```

As a result, all quotes will be in italics. Now suppose further that the user wants the quotes also in blue and therefore adds:

[4] there are simpler ways to achieve the same effect.

```
\usepackage{color}
\AddToHook{env/quote/before}{\begin{color}{blue}}
\AddToHook{env/quote/after} {\end{color}}
```

Now if the `env/quote/after` hook would be a normal hook we would get the same execution order in both hooks, namely:

```
package-1, top-level
```

(or vice versa) and as a result, would get:

```
\begin{itshape}\begin{color}{blue} ...
\end{itshape}\end{color}
```

and an error message that `\begin{color}` ended by `\end{itshape}`. With `env/quote/after` declared as a reversed hook the execution order is reversed and so all environments are closed in the correct sequence and `\ShowHook` would give us the following output:

```
The hook 'env/quote/after':
 Code chunks:
    package-1 -> \end {itshape}
    top-level -> \end {color}
 Extra code next invocation:
    ---
 Rules:
    ---
 Execution order (after reversal):
    top-level, package-1
```

The reversal of the execution order happens before applying any rules, so if you alter the order you will probably have to alter it in both hooks, not just in one, but that depends on the use case.

2.5 Difference between "normal" and "one-time" hooks

When executing a hook a developer has the choice of using either `\UseHook` or `\UseOneTimeHook` (or their `expl3` equivalents `\hook_use:n` and `\hook_use_once:n`). This choice affects how `\AddToHook` is handled after the hook has been executed for the first time.

With normal hooks adding code via `\AddToHook` means that the code chunk is added to the hook data structure and then used each time `\UseHook` is called.

With one-time hooks it this is handled slightly differently: After `\UseOneTimeHook` has been called, any further attempts to add code to the hook via `\AddToHook` will simply execute the ⟨*code*⟩ immediately.

This has some consequences one needs to be aware of:

- If ⟨*code*⟩ is added to a normal hook after the hook was executed and it is never executed again for one or the other reason, then this new ⟨*code*⟩ will never be executed.
- In contrast if that happens with a one-time hook the ⟨*code*⟩ is executed immediately.

In particular this means that construct such as

```
\AddToHook{myhook}
          { ⟨code-1⟩ \AddToHook{myhook}{⟨code-2⟩} ⟨code-3⟩ }
```

works for one-time hooks[5] (all three code chunks are executed one after another), but it makes little sense with a normal hook, because with a normal hook the first time `\UseHook{myhook}` is executed it would

- execute ⟨*code-1*⟩,
- then execute `\AddToHook{myhook}{code-2}` which adds the code chunk ⟨*code-2*⟩ to the hook for use on the next invocation,
- and finally execute ⟨*code-3*⟩.

The second time `\UseHook` is called it would execute the above and in addition ⟨*code-2*⟩ as that was added as a code chunk to the hook in the meantime. So each time the hook is used another copy of ⟨*code-2*⟩ is added and so that code chunk is executed ⟨*# of invocations*⟩ − 1 times.

2.6 Private LaTeX kernel hooks

There are a few places where it is absolutely essential for LaTeX to function correctly that code is executed in a precisely defined order. Even that could have been implemented with the hook management (by adding various rules to ensure the appropriate ordering with respect to other code added by packages). However, this makes every document unnecessary slow, because there has to be sorting even through the result is predetermined. Furthermore it forces package writers to unnecessarily add such rules if they add further code to the hook (or break LaTeX).

For that reason such code is not using the hook management, but instead private kernel commands directly before or after a public hook with the following naming convention: `\@kernel@before@`⟨*hookname*⟩ or `\@kernel@after@`⟨*hookname*⟩. For example, in `\enddocument` you find

```
\UseHook{enddocument}%
\@kernel@after@enddocument
```

which means first the user/package-accessible `enddocument` hook is executed and then the internal kernel hook. As their name indicates these kernel commands should not be altered by third-party packages, so please refrain from that in the interest of stability and instead use the public hook next to it.[6]

2.7 Legacy LaTeX 2ε interfaces

LaTeX 2ε offered a small number of hooks together with commands to add to them. They are listed here and are retained for backwards compatibility.

With the new hook management several additional hooks have been added to LaTeX and more will follow. See the next section for what is already available.

[5] This is sometimes used with `\AtBeginDocument` which is why it is supported.

[6] As with everything in TeX there is not enforcement of this rule, and by looking at the code it is easy to find out how the kernel adds to them. The main reason of this section is therefore to say "please don't do that, this is unconfigurable code!"

\AtBeginDocument

\AtBeginDocument [⟨*label*⟩] {⟨*code*⟩}

If used without the optional argument ⟨*label*⟩, it works essentially like before, i.e., it is adding ⟨*code*⟩ to the hook `begindocument` (which is executed inside `\begin{document}`). However, all code added this way is labeled with the label `top-level` if done outside of a package or class or with the package/class name if called inside such a file.

This way one can add further code to the hook using `\AddToHook` or `\AtBeginDocument` using a different label and explicitly order the code chunks as necessary, e.g., run some code before or after the `top-level` code. When using the optional argument the call is equivalent to running `\AddToHook {begindocument} [`⟨*label*⟩`] {`⟨*code*⟩`}`.

`\AtBeginDocument` is a wrapper around the `begindocument` hook (see section 2.8.2), which is a one-time hook. As such, after the `begindocument` hook is executed at `\begin{document}` any attempt to add ⟨*code*⟩ to this hook with `\AtBeginDocument` or with `\AddToHook` will cause that ⟨*code*⟩ to execute immediately instead. See section 2.5 for more on one-time hooks.

For important packages with known order requirement we may over time add rules to the kernel (or to those packages) so that they work regardless of the loading-order in the document.

\AtEndDocument

\AtEndDocument [⟨*label*⟩] {⟨*code*⟩}

Like `\AtBeginDocument` but for the `enddocument` hook.

\AtBeginDvi

\AtBeginDvi [⟨*label*⟩] {⟨*code*⟩}

This hook is discussed in conjunction with the shipout hooks.

The few hooks that existed previously in LaTeX 2ε used internally commands such as `\@begindocumenthook` and packages sometimes augemented them directly rather than working through `\AtBeginDocumement`. For that reason there is currently support for this, that is, if the system detects that such an internal legacy hook command contains code it adds it to the new hook system under the label `legacy` so that it doesn't get lost.

However, over time the remaining cases of direct usage need updating because in one of the future release of LaTeX we will turn this legacy support off, as it does unnecessary slow down the processing.

2.8 LaTeX 2ε commands and environments augmented by hooks

intro to be written

2.8.1 Generic hooks for all environments

Every environment ⟨*env*⟩ has now four associated hooks coming with it:

`env/`⟨*env*⟩`/before` This hook is executed as part of `\begin` as the very first action, in particular prior to starting the environment group. Its scope is therefore not restricted by the environment.

`env/`⟨*env*⟩`/begin` This hook is executed as part of `\begin` directly in front of the code specific to the environment start (e.g., the second argument of `\newenvironment`). Its scope is the environment body.

env/⟨*env*⟩/end This hook is executed as part of `\end` directly in front of the code specific to the end of the environment (e.g., the third argument of `\newenvironment`).

env/⟨*env*⟩/after This hook is executed as part of `\end` after the code specific to the environment end and after the environment group has ended. Its scope is therefore not restricted by the environment.

The hook is implemented as a reversed hook so if two packages add code to `env/⟨env⟩/before` and to `env/⟨env⟩/after` they can add surrounding environments and the order of closing them happens in the right sequence.

Generic environment hooks are never one-time hooks even with environments that are supposed to appear only once in a document.[7] In contrast to other hooks there is also no need to declare them using `\NewHook`.

The hooks are only executed if `\begin{⟨env⟩}` and `\end{⟨env⟩}` is used. If the environment code is executed via low-level calls to `\⟨env⟩` and `\end⟨env⟩` (e.g., to avoid the environment grouping) they are not available. If you want them available in code using this method, you would need to add them yourself, i.e., write something like

```
\UseHook{env/quote/before}\quote
    ...
\endquote\UseHook{env/quote/after}
```

to add the outer hooks, etc.

`\BeforeBeginEnvironment`

`\BeforeBeginEnvironment [⟨label⟩] {⟨code⟩}`

This declaration adds to the `env/⟨env⟩/before` hook using by default the current package or class name as a label or `top-level` if used in the document directly.

`\AtBeginEnvironment`

`\AtBeginEnvironment [⟨label⟩] {⟨code⟩}`

Like `\BeforeBeginEnvironment` but adds to the `env/⟨env⟩/begin` hook.

`\AtEndEnvironment`

`\AtEndEnvironment [⟨label⟩] {⟨code⟩}`

Like `\BeforeBeginEnvironment` but adds to the `env/⟨env⟩/end` hook.

`\AfterEndEnvironment`

`\AfterEndEnvironment [⟨label⟩] {⟨code⟩}`

Like `\BeforeBeginEnvironment` but adds to the `env/⟨env⟩/after` hook.

2.8.2 Hooks provided by `\begin{document}`

Until 2020 `\begin{document}` offered exactly one hook that one could add to using `\AtBeginDocument`. Experiences over the years have shown that this single hook in one place was not enough and as part of adding the general hook management system a number of additional hooks have been added at this point. The places for these hooks have been chosen to provide the same support as offered by external packages, such as etoolbox and others that augmented `\document` to gain better control.

Supported are now the following hooks (all of them one-time hooks):

[7] Thus if one adds code to such hooks after the environment has been processed, it will only be executed if the environment appears again and if that doesn't happen the code will never get executed.

begindocument/before This hook is executed at the very start of `\document`, one can think of it as a hook for code at the end of the preamble section and this is how it is used by etoolbox's `\AtEndPreamble`.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

begindocument This hook is added to when using `\AtBeginDocument` and it is executed after the `.aux` file as be read in and most initialization are done, so they can be altered and inspected by the hook code. It is followed by a small number of further initializations that shouldn't be altered and are therefore coming later.

The hook should not be used to add material for typesetting as we are still in LaTeX's initialization phase and not in the document body. If such material needs to be added to the document body use the next hook instead.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

begindocument/end This hook is executed at the end of the `\document` code in other words at the beginning of the document body. The only command that follows it is `\ignorespaces`.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

The generic hooks executed by `\begin` also exist, i.e., `env/document/before` and `env/document/begin`, but with this special environment it is better use the dedicated one-time hooks above.

2.8.3 Hooks provided by `\end{document}`

LaTeX 2_ε always provided `\AtEndDocument` to add code to the execution of `\end{document}` just in front of the code that is normally executed there. While this was a big improvement over the situation in LaTeX 2.09 it was not flexible enough for a number of use cases and so packages, such as etoolbox, atveryend and others patched `\enddocument` to add additional points where code could be hooked into.

Patching using packages is always problematical as leads to conflicts (code availability, ordering of patches, incompatible patches, etc.). For this reason a number of additional hooks have been added to the `\enddocument` code to allow packages to add code in various places in a controlled way without the need for overwriting or patching the core code.

Supported are now the following hooks (all of them one-time hooks):

enddocument The hook associated with `\AtEndDocument`. It is immediately called at the beginning of `\enddocument`.

When this hook is executed there may be still unprocessed material (e.g., floats on the deferlist) and the hook may add further material to be typeset. After it, `\clearpage` is called to ensure that all such material gets typeset. If there is nothing waiting the `\clearpage` has no effect.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

enddocument/afterlastpage As the name indicates this hook should not receive code that generates material for further pages. It is the right place to do some final housekeeping and possibly write out some information to the `.aux` file (which is still open at this point to receive data). It is also the correct place to set up any testing code to be run when the `.aux` file is re-read in the next step.

After this hook has been executed the `.aux` file is closed for writing and then read back in to do some tests (e.g., looking for missing references or duplicated labels, etc.).

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

enddocument/afteraux At this point, the `.aux` file has been reprocessed and so this is a possible place for final checks and display of information to the user. However, for the latter you might prefer the next hook, so that your information is displayed after the (possibly longish) list of files if that got requested via `\listfiles`.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

enddocument/info This hook is meant to receive code that write final information messages to the terminal. It follows immediately after the previous hook (so both could have been combined, but then packages adding further code would always need to also supply an explicit rule to specify where it should go.

This hook already contains some code added by the kernel (under the labels `kernel/filelist` and `kernel/warnings`), namely the list of files when `\listfiles` has been used and the warnings for duplicate labels, missing references, font substitutions etc.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).

enddocument/end Finally, this hook is executed just in front of the final call to `\@@end`.

This is a one-time hook, so after it is executed, all further attempts to add code to it will execute such code immediately (see section 2.5).is it even possible to add code after this one?

There is also the hook `shipout/lastpage`. This hook is executed as part of the last `\shipout` in the document to allow package to add final `\special`'s to that page. Where this hook is executed in relation to those from the above list can vary from document to document. Furthermore to determine correctly which of the `\shipout`s is the last one, LaTeX needs to be run several times, so initially it might get executed on the wrong page. See section 2.8.4 for where to find the details.

It is in also possible to use the generic `env/document/end` hook which is execuded by `\end`, i.e., just in front of the first hook above. Note however that the other generic `\end` environment hook, i.e., `env/document/after` will never get executed, because by that time LaTeX has finished the document processing.

2.8.4 Hooks provided `\shipout` operations

There are several hooks and mechanisms added to LaTeX's process of generating pages. These are documented in `ltshipout-doc.pdf` or with code in `ltshipout-code.pdf`.

2.8.5 Hooks provided by file loading operations

There are several hooks added to LaTeX's process of loading file via its high-level interfaces such as `\input`, `\include`, `\usepackage`, etc. These are documented in `ltfilehook-doc.pdf` or with code in `ltfilehook-code.pdf`.

2.8.6 Hooks provided in NFSS commands

In languages that need to support for more than one script in parallel (and thus several sets of fonts), e.g., Latin and Japanese fonts, NFSS font commands, such as `\sffamily`, need to switch both the Latin family to "Sans Serif" and in addition alter a second set of fonts.

To support this several NFSS have hooks in which such support can be added.

`rmfamily` After `\rmfamily` has done its initial checks and prepared a any font series update this hook is executed and only afterwards `\selectfont`.

`sffamily` Like the `rmfamily` hook but for the `\sffamily` command.

`ttfamily` Like the `rmfamily` hook but for the `\ttfamily` command.

`normalfont` The `\normalfont` command resets font encoding family series and shape to their document defaults. It then executes this hook and finally calls `\selectfont`.

`expand@font@defaults` The internal `\expand@font@defaults` command expands and saves the current defaults for the meta families (rm/sf/tt) and the meta series (bf/md). If the NFSS machinery has been augmented, e.g., for Chinese or Japanese fonts, then further defaults may need to be set at this point. This can be done in this hook which is executed at the end of this macro.

`bfseries/defaults`, `bfseries` If the `\bfdefault` was explicitly changed by the user its new value is used to set the bf series defaults for the meta families (rm/sf/tt) when `\bfseries` is called. In the `bfseries/defaults` hook further adjustments can be made in this case. This hook is only executed if such a change is detected. In contrast the `bfseries` hook is always executed just before `\selectfont` is called to change to the new series.

`mdseries/defaults`, `mdseries` These two hooks are like the previous ones but used in `\mdseries` command.

3 The Implementation

3.1 Loading further extensions

```
⟨@@=hook⟩
```

At the moment the whole module rolls back in one go, but if we make any modifications in later releases this will then need splitting.

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\NewHook}{The hook management}%
\ExplSyntaxOn
```

3.2 Debugging

\g__hook_debug_bool

Holds the current debugging state.

```
\bool_new:N \g__hook_debug_bool
```

(*End definition for* \g__hook_debug_bool.)

\hook_debug_on:
\hook_debug_off:
__hook_debug:n
__hook_debug_gset:

Turns debugging on and off by redefining __hook_debug:n.

```
\cs_new_eq:NN \__hook_debug:n \use_none:n
\cs_new_protected:Npn \hook_debug_on:
  {
    \bool_gset_true:N \g__hook_debug_bool
    \__hook_debug_gset:
  }
\cs_new_protected:Npn \hook_debug_off:
  {
    \bool_gset_false:N \g__hook_debug_bool
    \__hook_debug_gset:
  }
\cs_new_protected:Npn \__hook_debug_gset:
  {
    \cs_gset_protected:Npx \__hook_debug:n ##1
      { \bool_if:NT \g__hook_debug_bool {##1} }
  }
```

(*End definition for* \hook_debug_on: *and others. These functions are documented on page* 111.)

3.3 Borrowing from internals of other kernel modules

__hook_str_compare:nn

Private copy of __str_if_eq:nn

```
\cs_new_eq:NN \__hook_str_compare:nn \__str_if_eq:nn
```

(*End definition for* __hook_str_compare:nn.)

3.4 Declarations

\l__hook_tmpa_bool

Scratch boolean used throughout the package.

```
\bool_new:N \l__hook_tmpa_bool
```

(*End definition for* \l__hook_tmpa_bool.)

\l__hook_return_tl
\l__hook_tmpa_tl
\l__hook_tmpb_tl

Scratch variables used throughout the package.

```
\tl_new:N \l__hook_return_tl
\tl_new:N \l__hook_tmpa_tl
\tl_new:N \l__hook_tmpb_tl
```

(*End definition for* \l__hook_return_tl, \l__hook_tmpa_tl, *and* \l__hook_tmpb_tl.)

\g__hook_all_seq

In a few places we need a list of all hook names ever defined so we keep track if them in this sequence.

```
\seq_new:N \g__hook_all_seq
```

(*End definition for* \g__hook_all_seq.)

\g__hook_removal_list_prop A token list to hold delayed removals.

```
\tl_new:N \g__hook_removal_list_tl
```

(*End definition for* \g__hook_removal_list_prop.)

\l__hook_cur_hook_tl Stores the name of the hook currently being sorted.

```
\tl_new:N \l__hook_cur_hook_tl
```

(*End definition for* \l__hook_cur_hook_tl.)

\l__hook_work_prop A property list holding a copy of the \g__hook_⟨*hook*⟩_code_prop of the hook being sorted to work on, so that changes don't act destructively on the hook data structure.

```
\prop_new:N \l__hook_work_prop
```

(*End definition for* \l__hook_work_prop.)

\g__hook_execute_immediately_prop List of hooks that from no on should not longer receive code.

```
\prop_new:N \g__hook_execute_immediately_prop
```

(*End definition for* \g__hook_execute_immediately_prop.)

\g__hook_used_prop All hooks that receive code (for use in debugging display).

```
\prop_new:N \g__hook_used_prop
```

(*End definition for* \g__hook_used_prop.)

\g__hook_hook_curr_name_tl
\g__hook_name_stack_seq Default label used for hook commands, and a stack to keep track of packages within packages.

```
\tl_new:N \g__hook_hook_curr_name_tl
\seq_new:N \g__hook_name_stack_seq
```

(*End definition for* \g__hook_hook_curr_name_tl *and* \g__hook_name_stack_seq.)

__hook_tmp:w Temporary macro for generic usage.

```
\cs_new_eq:NN \__hook_tmp:w ?
```

(*End definition for* __hook_tmp:w.)

\tl_gremove_once:Nx
\tl_show:x
\tl_log:x Some variants of expl3 functions.

FMi: should be moved to expl3

```
\cs_generate_variant:Nn \tl_gremove_once:Nn { Nx }
\cs_generate_variant:Nn \tl_show:n { x }
\cs_generate_variant:Nn \tl_log:n { x }
```

(*End definition for* \tl_gremove_once:Nx, \tl_show:x, *and* \tl_log:x. *These functions are documented on page* ??.)

\s__hook_mark Scan mark used for delimited arguments.

```
\scan_new:N \s__hook_mark
```

(*End definition for* \s__hook_mark.)

3.5 Providing new hooks

\g__hook_..._code_prop
__hook~...
__hook_next~...

Hooks have a ⟨*name*⟩ and for each hook we have to provide a number of data structures. These are

\g__hook_⟨*name*⟩_code_prop A property list holding the code for the hook in separate chunks. The keys are by default the package names that add code to the hook, but it is possible for packages to define other keys.

\g__hook_⟨*name*⟩_rule_⟨*label1*⟩|⟨*label2*⟩_tl A token list holding the relation between ⟨*label1*⟩ and ⟨*label2*⟩ in the ⟨*name*⟩. The ⟨*labels*⟩ are lexically (reverse) sorted to ensure that two labels always point to the same token list. For global rules, the ⟨*name*⟩ is ??.

__hook ⟨*name*⟩ The code that is actually executed when the hook is called in the document is stored in this token list. It is constructed from the code chunks applying the information. This token list is named like that so that in case of an error inside the hook, the reported token list in the error is shorter, and to make it simpler to normalize hook names in __hook_make_name:n.

\g__hook_⟨*name*⟩_reversed_tl Some hooks are "reversed". This token list stores a - for such hook so that it can be identified. The - character is used because ⟨*reversed*⟩1 is +1 for normal hooks and −1 for reversed ones.

__hook_next ⟨*name*⟩ Finally there is extra code (normally empty) that is used on the next invocation of the hook (and then deleted). This can be used to define some special behavior for a single occasion from within the document. This token list follows the same naming scheme than the main __hook ⟨*name*⟩ token list. It is called __hook_next ⟨*name*⟩ rather than __hook next_⟨*name*⟩ because otherwise a hook whose name is next_⟨*name*⟩ would clash with the next code-token list of the hook called ⟨*name*⟩.

(*End definition for* \g__hook_..._code_prop, __hook~..., *and* __hook_next~....)

\hook_new:n

The \hook_new:n declaration declare a new hook and expects the hook ⟨*name*⟩ as its argument, e.g., begindocument.

```
\cs_new_protected:Npn \hook_new:n #1
  { \__hook_normalize_hook_args:Nn \__hook_new:n {#1} }
\cs_new_protected:Npn \__hook_new:n #1
  {
```

We check for one of the internal data structures and if it already exists we complain.

```
    \hook_if_exist:nTF {#1}
      { \msg_error:nnn { hooks } { exists } {#1} }
```

Otherwise we add the hook name to the list of all hooks and allocate the necessary data structures for the new hook.

```
      {
        \seq_gput_right:Nn \g__hook_all_seq {#1}
```

This is only used by the actual code of the current hook, so declare it normally:

```
        \tl_new:c { __hook~#1 }
```

Now ensure that the base data structure for the hook exists:

```
        \__hook_declare:n {#1}
```

The `\g__hook_⟨hook⟩_labels_clist` holds the sorted list of labels (once it got sorted). This is used only for debugging.

```
        \clist_new:c {g__hook_#1_labels_clist}
```

Some hooks should reverse the default order of code chunks. To signal this we have a token list which is empty for normal hooks and contains a `-` for reversed hooks.

```
        \tl_new:c { g__hook_#1_reversed_tl }
```

The above is all in L3 convention, but we also provide an interface to legacy LaTeX 2ε hooks of the form `\@...hook`, e.g., `\@begindocumenthook`. there have been a few of them and they have been added to using `\g@addto@macro`. If there exists such a macro matching the name of the new hook, i.e., `\@⟨hook-name⟩hook` and it is not empty then we add its contents as a code chunk under the label `legacy`.

Warning: this support will vanish in future releases!

```
        \__hook_include_legacy_code_chunk:n {#1}
    }
}
```

(*End definition for* `\hook_new:n`*. This function is documented on page 109.*)

`\__hook_declare:n` This function declares the basic data structures for a hook without actually declaring the hook itself. This is needed to allow adding to undeclared hooks. Here it is unnecessary to check whether both variables exist, since both are declared at the same time (either both exist, or neither).

```
\cs_new_protected:Npn \__hook_declare:n #1
  {
    \__hook_if_exist:nF {#1}
      {
        \prop_new:c { g__hook_#1_code_prop }
        \tl_new:c { __hook_next~#1 }
      }
  }
```

(*End definition for* `\__hook_declare:n`*.*)

`\hook_new_reversed:n` `\__hook_new_reversed:n` Declare a new hook. The default ordering of code chunks is reversed, signaled by setting the token list to a minus sign.

```
\cs_new_protected:Npn \hook_new_reversed:n #1
  { \__hook_normalize_hook_args:Nn \__hook_new_reversed:n {#1} }
\cs_new_protected:Npn \__hook_new_reversed:n #1
  {
    \__hook_new:n {#1}
```

If the hook already exists the above will generate an error message, so the next line should be executed (but it is — too bad).

```
    \tl_gset:cn { g__hook_#1_reversed_tl } { - }
  }
```

(*End definition for* `\hook_new_reversed:n` *and* `\__hook_new_reversed:n`*. This function is documented on page 109.*)

`\hook_new_pair:nn` A shorthand for declaring a normal and a (matching) reversed hook in one go.

```
\cs_new_protected:Npn \hook_new_pair:nn #1#2
  { \hook_new:n {#1} \hook_new_reversed:n {#2} }
```

(*End definition for* `\hook_new_pair:nn`. *This function is documented on page* 109.)

`\__hook_include_legacy_code_chunk:n`

The LaTeX legacy concept for hooks uses with hooks the following naming scheme in the code: `\@...hook`.

If this macro is not empty we add it under the label `legacy` to the current hook and then empty it globally. This way packages or classes directly manipulating commands such as `\@begindocumenthook` still get their hook data added.

Warning: this support will vanish in future releases!

```
\cs_new_protected:Npn \__hook_include_legacy_code_chunk:n #1
  {
```

If the expl3 code is run with checking on then assigning or using non L3 names such as `\@enddocumenthook` with expl3 functions will trigger warnings so we run this code with debugging explicitly suspended.

```
    \debug_suspend:
```

If the macro doesn't exist (which is the usual case) then nothing needs to be done.

```
    \tl_if_exist:cT { @#1hook }
```

Of course if the legacy hook exists but is empty, there is no need to add anything under `legacy` the legacy label.

```
      {
        \tl_if_empty:cF { @#1hook }
          {
            \exp_args:Nnnv \__hook_hook_gput_code_do:nnn {#1}
                                    { legacy } { @#1hook }
```

Once added to the hook, we need to clear it otherwise it might get added again later if the hook data gets updated.

```
            \tl_gclear:c { @#1hook }
          }
      }
    \debug_resume:
  }
```

(*End definition for* `\__hook_include_legacy_code_chunk:n`.)

3.6 Parsing a label

`\__hook_parse_label_default:n`

This macro checks if a label was given (not `\c_novalue_tl`), and if so, tries to parse the label looking for a leading `.` to replace for `\@currname`. Otherwise `\__hook_currname_or_default:n` is used to pick `\@currname` or the fallback value.

The default `top-level` is hard-coded here. It once was an argument, but it's no longer needed.

PhO: can't remember why

```
\cs_new:Npn \__hook_parse_label_default:n #1
  {
    \tl_if_novalue:nTF {#1}
      { \__hook_currname_or_default:n { top-level } }
      {
        \tl_trim_spaces_apply:nN {#1}
          \__hook_parse_dot_label:nn { top-level }
      }
  }
```

(*End definition for* `\__hook_parse_label_default:n`.)

`\__hook_parse_dot_label:nn`
`\__hook_parse_dot_label:nw`
`\__hook_parse_dot_label_cleanup:w`
`\__hook_parse_dot_label_aux:nw`

Start by checking if the label is empty, which raises an error, and uses the fallback value. If not, split the label at a `./`, if any, and check if no tokens are before the `./`, or if the only character is a `.`. If these requirements are fulfilled, the leading `.` is replaced with `\__hook_currname_or_default:n`. Otherwise the label is returned unchanged.

```
\cs_new:Npn \__hook_parse_dot_label:nn #1 #2
  {
    \tl_if_empty:nTF {#1}
      {
        \msg_expandable_error:nnn { hooks } { empty-label } {#2}
        #2
      }
      {
        \str_if_eq:nnTF {#1} { . }
          { \__hook_currname_or_default:n {#2} }
          { \__hook_parse_dot_label:nw {#2} #1 ./ \s__hook_mark }
      }
  }
\cs_new:Npn \__hook_parse_dot_label:nw #1 #2 ./ #3 \s__hook_mark
  {
    \tl_if_empty:nTF {#2}
      { \__hook_parse_dot_label_aux:nw {#1} #3 \s__hook_mark }
      {
        \tl_if_empty:nTF {#3}
          { \__hook_make_name:n {#2} }
          { \__hook_parse_dot_label_cleanup:w #2 ./ #3 \s__hook_mark }
      }
  }
\cs_new:Npn \__hook_parse_dot_label_cleanup:w #1 ./ \s__hook_mark {#1}
\cs_new:Npn \__hook_parse_dot_label_aux:nw #1 #2 ./ \s__hook_mark
  { \__hook_currname_or_default:n {#1} / \__hook_make_name:n {#2} }
```

(*End definition for* `\__hook_parse_dot_label:nn` *and others.*)

`\__hook_currname_or_default:n`

Uses `\g__hook_hook_curr_name_tl` if it is set, otherwise tries `\@currname`. If neither is set, uses the fallback value `#1` (usually `top-level`).

```
\cs_new:Npn \__hook_currname_or_default:n #1
  {
    \tl_if_empty:NTF \g__hook_hook_curr_name_tl
      {
        \tl_if_empty:NTF \@currname
          { \__hook_make_name:n {#1} }
          { \@currname }
      }
      { \g__hook_hook_curr_name_tl }
  }
```

(*End definition for* `\__hook_currname_or_default:n`.)

`\__hook_make_name:n`
`\__hook_make_name:w`

Provides a standard sanitisation of a hook's name. It uses `\cs:w` to build a control sequence out of the hook name, then uses `\cs_to_str:N` to get the string representation of that, without the escape character. `\cs:w`-based expansion is used instead of `e`-based because Unicode characters don't behave well inside `\expanded`. The macro adds the

`\__hook~` prefix to the hook name to reuse the hook's code token list to build the csname and avoid leaving "public" control sequences defined (as `\relax`) in TeX's memory.

```
\cs_new:Npn \__hook_make_name:n #1
  {
    \exp_after:wN \exp_after:wN \exp_after:wN \__hook_make_name:w
    \exp_after:wN \token_to_str:N \cs:w __hook~ #1 \cs_end:
  }
\exp_last_unbraced:NNNNo
\cs_new:Npn \__hook_make_name:w #1 \tl_to_str:n { __hook~ } { }
```

(*End definition for* `\__hook_make_name:n` *and* `\__hook_make_name:w`.)

`\__hook_normalize_hook_args:Nn`
`\__hook_normalize_hook_args:Nnn`
`\__hook_normalize_hook_rule_args:Nnnnn`
`\__hook_normalize_hook_args_aux:Nn`

Standard route for normalising hook and label arguments. The main macro does the entire operation within a group so that csnames made by `\__hook_make_name:n` are wiped off before continuing. This means that this function cannot be used for `\hook_use:n`!

```
\cs_new_protected:Npn \__hook_normalize_hook_args_aux:Nn #1 #2
  {
    \group_begin:
    \use:e
      {
        \group_end:
        \exp_not:N #1 #2
      }
  }
\cs_new_protected:Npn \__hook_normalize_hook_args:Nn #1 #2
  {
    \__hook_normalize_hook_args_aux:Nn #1
      { { \__hook_parse_label_default:n {#2} } }
  }
\cs_new_protected:Npn \__hook_normalize_hook_args:Nnn #1 #2 #3
  {
    \__hook_normalize_hook_args_aux:Nn #1
      {
        { \__hook_parse_label_default:n {#2} }
        { \__hook_parse_label_default:n {#3} }
      }
  }
\cs_new_protected:Npn \__hook_normalize_hook_rule_args:Nnnnn #1 #2 #3 #4 #5
  {
    \__hook_normalize_hook_args_aux:Nn #1
      {
        { \__hook_parse_label_default:n {#2} }
        { \__hook_parse_label_default:n {#3} }
        { \tl_trim_spaces:n {#4} }
        { \__hook_parse_label_default:n {#5} }
      }
  }
```

(*End definition for* `\__hook_normalize_hook_args:Nn` *and others.*)

`\hook_gput_code:nnn`
`\__hook_gput_code:nnn`
`\__hook_gput_code:nxv`
`\__hook_hook_gput_code_do:nnn`

With `\hook_gput_code:nnn{`⟨*hook*⟩`}{`⟨*label*⟩`}{`⟨*code*⟩`}` a chunk of ⟨*code*⟩ is added to an existing ⟨*hook*⟩ labeled with ⟨*label*⟩.

```
\cs_new_protected:Npn \hook_gput_code:nnn #1 #2
```

```
  { \__hook_normalize_hook_args:Nnn \__hook_gput_code:nnn {#1} {#2} }
\cs_new_protected:Npn \__hook_gput_code:nnn #1 #2 #3
  {
```

First check if the hook was used as a one-time hook:

```
    \prop_if_in:NnTF \g__hook_execute_immediately_prop {#1}
      {#3}
      {
```

Then check if the current ⟨*hook*⟩/⟨*label*⟩ pair was marked for removal, in which case `\__hook_unmark_removal:nn` is used to remove that mark (once). This may happen when a package removes code from another package which was not yet loaded: the removal order is stored, and at this stage it is executed by not adding to the hook.

```
        \__hook_if_marked_removal:nnTF {#1} {#2}
          { \__hook_unmark_removal:nn {#1} {#2} }
          {
```

If no removal is queued, we are free to add. Start by checking if the hook exists.

```
            \hook_if_exist:nTF {#1}
```

If so we simply add (or append) the new code to the property list holding different chunks for the hook. At `\begin{document}` this is then sorted into a token list for fast execution.

```
              {
                \__hook_hook_gput_code_do:nnn {#1} {#2} {#3}
```

However, if there is an update within the document we need to alter this execution code which is done by `\__hook_update_hook_code:n`. In the preamble this does nothing.

```
                \__hook_update_hook_code:n {#1}
              }
```

If the hook does not exist, however, before giving up try to declare it as a generic hook, if its name matches one of the valid patterns.

```
              { \__hook_try_declaring_generic_hook:nnn {#1} {#2} {#3} }
          }
      }
  }
\cs_generate_variant:Nn \__hook_gput_code:nnn { nxv }
```

This macro will unconditionally add a chunk of code to the given hook.

```
\cs_new_protected:Npn \__hook_hook_gput_code_do:nnn #1 #2 #3
  {
```

However, first some debugging info if debugging is enabled:

```
    \__hook_debug:n{\iow_term:x{****~ Add~ to~
                        \hook_if_exist:nF {#1} { undeclared~ }
                        hook~ #1~ (#2)
                        \on@line\space <-~ \tl_to_str:n{#3}} }
```

Then try to get the code chunk labeled `#2` from the hook. If there's code already there, then append `#3` to that, otherwise just put `#3`.

```
    \prop_get:cnNTF { g__hook_#1_code_prop } {#2} \l__hook_return_tl
      {
        \prop_gput:cno { g__hook_#1_code_prop } {#2}
          { \l__hook_return_tl #3 }
      }
      { \prop_gput:cnn { g__hook_#1_code_prop } {#2} {#3} }
  }
```

(*End definition for* `\hook_gput_code:nnn`, `\__hook_gput_code:nnn`, *and* `\__hook_hook_gput_code_do:nnn`. *This function is documented on page* 109.)

__hook_gput_undeclared_hook:nnn

Often it may happen that a package A defines a hook `foo`, but package B, that adds code to that hook, is loaded before A. In such case we need to add code to the hook before its declared.

```
\cs_new_protected:Npn \__hook_gput_undeclared_hook:nnn #1 #2 #3
  {
    \__hook_declare:n {#1}
    \__hook_hook_gput_code_do:nnn {#1} {#2} {#3}
  }
```

(*End definition for* `\__hook_gput_undeclared_hook:nnn`.)

__hook_try_declaring_generic_hook:nnn
__hook_try_declaring_generic_next_hook:nn

These entry-level macros just pass the arguments along to the common `\__hook_try_declaring_generic_hook:nNNnn` with the right functions to execute when some action is to be taken.

The wrapper `\__hook_try_declaring_generic_hook:nnn` then defers `\hook_gput_code:nnn` if the generic hook was declared, or to `\__hook_gput_undeclared_hook:nnn` otherwise (the hook was tested for existence before, so at this point if it isn't generic, it doesn't exist).

The wrapper `\__hook_try_declaring_generic_next_hook:nn` for next-execution hooks does the same: it defers the code to `\hook_gput_next_code:nn` if the generic hook was declared, or to `\__hook_gput_next_do:nn` otherwise.

```
\cs_new_protected:Npn \__hook_try_declaring_generic_hook:nnn #1
  {
    \__hook_try_declaring_generic_hook:nNNnn {#1}
      \hook_gput_code:nnn \__hook_gput_undeclared_hook:nnn
  }
\cs_new_protected:Npn \__hook_try_declaring_generic_next_hook:nn #1
  {
    \__hook_try_declaring_generic_hook:nNNnn {#1}
      \hook_gput_next_code:nn \__hook_gput_next_do:nn
  }
```

__hook_try_declaring_generic_hook:nNNnn
hook_try_declaring_generic_hook_split:nNNnn
__hook_try_declaring_generic_hook:wn*TF*

`\__hook_try_declaring_generic_hook:nNNnn` now splits the hook name at the first `/` (if any) and first checks if it is a file-specific hook (they require some normalization) using `\__hook_if_file_hook:wTF`. If not then check it is one of a predefined set for generic names. We also split off the second component to see if we have to make a reversed hook. In either case the function returns ⟨*true*⟩ for a generic hook and ⟨*false*⟩ in other cases.

```
\cs_new_protected:Npn \__hook_try_declaring_generic_hook:nNNnn #1
  {
    \__hook_if_file_hook:wTF #1 / / \s__hook_mark
      {
        \exp_args:Ne \__hook_try_declaring_generic_hook_split:nNNnn
          { \exp_args:Ne \__hook_file_hook_normalize:n {#1} }
      }
      { \__hook_try_declaring_generic_hook_split:nNNnn {#1} }
  }
\cs_new_protected:Npn \__hook_try_declaring_generic_hook_split:nNNnn #1 #2 #3
  {
    \__hook_try_declaring_generic_hook:wnTF #1 / / / \scan_stop: {#1}
      { #2 }
```

```
      { #3 } {#1}
  }
\prg_new_protected_conditional:Npnn \__hook_try_declaring_generic_hook:wn
    #1 / #2 / #3 / #4 \scan_stop: #5 { TF }
  {
    \tl_if_empty:nTF {#2}
      { \prg_return_false: }
      {
        \prop_if_in:NnTF \c__hook_generics_prop {#1}
          {
            \hook_if_exist:nF {#5} { \hook_new:n {#5} }
```

After having declared the hook we check the second component (for file hooks) or the third component for environment hooks) and if it is on the list of components for which we should have declared a reversed hook we alter the hook data structure accordingly.

```
            \prop_if_in:NnTF \c__hook_generics_reversed_ii_prop {#2}
              { \tl_gset:cn { g__hook_#5_reversed_tl } { - } }
              {
                \prop_if_in:NnT \c__hook_generics_reversed_iii_prop {#3}
                  { \tl_gset:cn { g__hook_#5_reversed_tl } { - } }
              }
```

Now that we know that the hook is declared we can add the code to it.

```
            \prg_return_true:
          }
          { \prg_return_false: }
      }
  }
```

(*End definition for* `\__hook_try_declaring_generic_hook:nnn` *and others.*)

`\__hook_if_file_hook_p:w`
`\__hook_if_file_hook:wTF`

`\__hook_if_file_hook:wTF` checks if the argument is a valid file-specific hook (not, for example, `file/before`, but `file/before/foo.tex`). If it is a file-specific hook, then it executes the ⟨*true*⟩ branch, otherwise ⟨*false*⟩.

A file-specific hook is `file/`⟨*position*⟩`/`⟨*name*⟩. If any of these parts don't exist, it is a general file hook or not a file hook at all, so the conditional evaluates to ⟨*false*⟩. Otherwise, it checks that the first part is `file` and that the ⟨*position*⟩ is in the `\c__hook_generics_file_prop`.

A property list is used here to avoid having to worry with catcodes, because expl3's file name parsing turns all characters into catcode-12 tokens, which might differ from hand-input letters.

```
\prg_new_conditional:Npnn \__hook_if_file_hook:w
    #1 / #2 / #3 \s__hook_mark { TF }
  {
    \str_if_eq:nnTF {#1} { file }
      {
        \bool_lazy_or:nnTF
            { \tl_if_empty_p:n {#3} }
            { \str_if_eq_p:nn {#3} { / } }
          { \prg_return_false: }
          {
            \prop_if_in:NnTF \c__hook_generics_file_prop {#2}
              { \prg_return_true: }
              { \prg_return_false: }
```

```
            }
        }
        { \prg_return_false: }
    }
```

(*End definition for* `\__hook_if_file_hook:wTF`.)

`\__hook_file_hook_normalize:n`
`\__hook_strip_double_slash:n`
`\__hook_strip_double_slash:w`

When a file-specific hook is found, before being declared it is lightly normalized by `\__hook_file_hook_normalize:n`. The current implementation just replaces two consecutive slashes (`//`) by a single one, to cope with simple cases where the user did something like `\def\input@path{{./mypath/}}`, in which case a hook would have to be `\AddToHook{file/after/./mypath//file.tex}`.

```
\cs_new:Npn \__hook_file_hook_normalize:n #1
  { \__hook_strip_double_slash:n {#1} }
\cs_new:Npn \__hook_strip_double_slash:n #1
  { \__hook_strip_double_slash:w #1 // \s__hook_mark }
\cs_new:Npn \__hook_strip_double_slash:w #1 // #2 \s__hook_mark
  {
    \tl_if_empty:nTF {#2}
      {#1}
      { \__hook_strip_double_slash:w #1 / #2 \s__hook_mark }
  }
```

(*End definition for* `\__hook_file_hook_normalize:n`, `\__hook_strip_double_slash:n`, *and* `\__hook_strip_double_slash:w`.)

`\c__hook_generics_prop`

Property list holding the generic names. We don't provide any user interface to this as this is meant to be static.

`env` The generic hooks used in `\begin` and `\end`.

`file` The generic hooks used when loading a file

```
\prop_const_from_keyval:Nn \c__hook_generics_prop
  {env=,file=,package=,class=,include=}
```

(*End definition for* `\c__hook_generics_prop`.)

`\c__hook_generics_reversed_ii_prop`
`\c__hook_generics_reversed_iii_prop`
`\c__hook_generics_file_prop`

Some of the generic hooks are supposed to use reverse ordering, these are the following (only the second or third sub-component is checked):

```
\prop_const_from_keyval:Nn \c__hook_generics_reversed_ii_prop {after=,end=}
\prop_const_from_keyval:Nn \c__hook_generics_reversed_iii_prop {after=}
\prop_const_from_keyval:Nn \c__hook_generics_file_prop {before=,after=}
```

(*End definition for* `\c__hook_generics_reversed_ii_prop`, `\c__hook_generics_reversed_iii_prop`, *and* `\c__hook_generics_file_prop`.)

`\hook_gremove_code:nn`
`\__hook_gremove_code:nn`

With `\hook_gremove_code:nn`{⟨*hook*⟩}{⟨*label*⟩} any code for ⟨*hook*⟩ stored under ⟨*label*⟩ is removed.

```
\cs_new_protected:Npn \hook_gremove_code:nn #1 #2
  { \__hook_normalize_hook_args:Nnn \__hook_gremove_code:nn {#1} {#2} }
\cs_new_protected:Npn \__hook_gremove_code:nn #1 #2
  {
```

First check that the hook code pool exists. `\hook_if_exist:nTF` isn't used here because it should be possible to remove code from a hook before its defined (see section 2.1.5).

```
    \__hook_if_exist:nTF {#1}
```

Then remove the chunk and run `\__hook_update_hook_code:n` so that the execution token list reflects the change if we are after `\begin{document}`.

```
      {
        \str_if_eq:nnTF {#2} {*}
          { \prop_gclear:c { g__hook_#1_code_prop } }
          {
```

Check if the label being removed exists in the code pool. If it does, just call `\__hook_gremove_code_do:nn` to do the removal, otherwise mark it to be removed.

```
            \prop_get:cnNTF { g__hook_#1_code_prop } {#2} \l__hook_return_tl
              { \__hook_gremove_code_do:nn }
              { \__hook_mark_removal:nn }
                  {#1} {#2}
          }
```

Finally update the code, if the hook exists.

```
        \hook_if_exist:nT {#1}
          { \__hook_update_hook_code:n {#1} }
      }
```

If the code pool for this hook doesn't exist it means that nothing tried to add to it before, so we just queue this removal order for later.

```
      { \__hook_mark_removal:nn {#1} {#2} }
  }
```

`\__hook_gremove_code_do:nn`

```
\cs_new_protected:Npn \__hook_gremove_code_do:nn #1 #2
  { \prop_gremove:cn { g__hook_#1_code_prop } {#2} }
```

(*End definition for* `\hook_gremove_code:nn`*,* `\__hook_gremove_code:nn`*, and* `\__hook_gremove_code_do:nn`*. This function is documented on page 110.*)

`\__hook_mark_removal:nn`

Marks ⟨*label*⟩ (`#2`) to be removed from ⟨*hook*⟩ (`#1`). The number of removals should be fairly small, and `\tl_gremove_once:Nx` is fairly efficient even for longer token lists, so we use a single global token list, rather than one for each hook.

A hand-crafted token list is used here because property lists don't hold repeated items, so multiple usages of `\__hook_mark_removal:nn` would be cancelled by a single `\__hook_unmark_removal:nn`.

```
\cs_new_protected:Npn \__hook_mark_removal:nn #1 #2
  {
    \tl_gput_right:Nx \g__hook_removal_list_tl
      { \__hook_removal_tl:nn {#1} {#2} }
  }
```

(*End definition for* `\__hook_mark_removal:nn`.)

`\__hook_unmark_removal:nn`

Unmarks ⟨*label*⟩ (`#2`) to be removed from ⟨*hook*⟩ (`#1`). `\tl_gremove_once:Nx` is used rather than `\tl_gremove_all:Nx` so that two additions are needed to cancel two marked removals, rather than only one.

```
\cs_new_protected:Npn \__hook_unmark_removal:nn #1 #2
  {
    \tl_gremove_once:Nx \g__hook_removal_list_tl
      { \__hook_removal_tl:nn {#1} {#2} }
  }
```

(*End definition for* `\__hook_unmark_removal:nn`.)

`__hook_if_marked_removal:nnTF` Checks if the `\g__hook_removal_list_tl` contains the current ⟨*label*⟩ (`#2`) and ⟨*hook*⟩ (`#1`).

```
\prg_new_protected_conditional:Npnn \__hook_if_marked_removal:nn #1 #2 { TF }
  {
    \exp_args:NNx \tl_if_in:NnTF \g__hook_removal_list_tl
      { \__hook_removal_tl:nn {#1} {#2} }
      { \prg_return_true: } { \prg_return_false: }
  }
```

(*End definition for* `\__hook_if_marked_removal:nnTF`.)

`\__hook_removal_tl:nn` Builds a token list with `#1` and `#2` which can only be matched by `#1` and `#2`. The `&`$_4$ anchors a removal, so that `#1` can't be mistaken by `#2` and vice versa, and the two `$`$_3$ delimit the two arguments

```
\cs_new:Npn \__hook_removal_tl:nn #1 #2
  { & \tl_to_str:n {#2} $ \tl_to_str:n {#1} $ }
```

(*End definition for* `\__hook_removal_tl:nn`.)

`\g__hook_??_code_prop` `\__hook~??` `\g__hook_??_reversed_tl` Initially these variables simply used an empty "label" name (not two question marks). This was a bit unfortunate, because then `l3doc` complains about `__` in the middle of a command name when trying to typeset the documentation. However using a "normal" name such as `default` has the disadvantage of that being not really distinguishable from a real hook name. I now have settled for `??` which needs some gymnastics to get it into the csname, but since this is used a lot things should be fast, so this is not done with `c` expansion in the code later on.

`\__hook~??` isn't used, but it has to be defined to trick the code into thinking that `??` is actually a hook.

```
\prop_new:c {g__hook_??_code_prop}
\prop_new:c {__hook~??}
```

Default rules are always given in normal ordering (never in reversed ordering). If such a rule is applied to a reversed hook it behaves as if the rule is reversed (e.g., `after` becomes `before`) because those rules are applied first and then the order is reversed.

```
\tl_new:c {g__hook_??_reversed_tl}
```

(*End definition for* `\g__hook_??_code_prop`, `\__hook~??`, *and* `\g__hook_??_reversed_tl`.)

3.7 Setting rules for hooks code

`\hook_gset_rule:nnnn` `\__hook_gset_rule:nnnn` *FMi: needs docu correction given new implementation*

With `\hook_gset_rule:nnnn`{⟨*hook*⟩}{⟨*label1*⟩}{⟨*relation*⟩}{⟨*label2*⟩} a relation is defined between the two code labels for the given ⟨*hook*⟩. The special hook `??` stands for *any* hook describing a default rule.

```
\cs_new_protected:Npn \hook_gset_rule:nnnn #1#2#3#4
  {
    \__hook_normalize_hook_rule_args:Nnnnn \__hook_gset_rule:nnnn
      {#1} {#2} {#3} {#4}
  }
```

```
\cs_new_protected:Npn \__hook_gset_rule:nnnn #1#2#3#4
  {
```

First we ensure the basic data structure of the hook exists:

```
    \__hook_declare:n {#1}
```

Then we clear any previous relationship between both labels.

```
    \__hook_rule_gclear:nnn {#1} {#2} {#4}
```

Then we call the function to handle the given rule. Throw an error if the rule is invalid.

```
    \debug_suspend:
    \cs_if_exist_use:cTF { __hook_rule_#3_gset:nnn }
      {
          {#1} {#2} {#4}
        \__hook_update_hook_code:n {#1}
      }
      { \msg_error:nnnnnn { hooks } { unknown-rule }
                          {#1} {#2} {#3} {#4}        }
    \debug_resume:
  }
```

(*End definition for* `\hook_gset_rule:nnnn` *and* `\__hook_gset_rule:nnnn`*. This function is documented on page 110.*)

`\__hook_rule_before_gset:nnn`
`\__hook_rule_after_gset:nnn`
`\__hook_rule_<_gset:nnn`
`\__hook_rule_>_gset:nnn`

Then we add the new rule. We need to normalize the rules here to allow for faster processing later. Given a pair of labels l_A and l_B, the rule $l_A > l_B$ is the same as $l_B < l_A$ only presented differently. But by normalizing the forms of the rule to a single representation, say, $l_B < l_A$, reduces the time spent looking for the rules later considerably.

Here we do that normalization by using `\(pdf)strcmp` to lexically sort labels l_A and l_B to a fixed order. This order is then enforced every time these two labels are used together.

Here we use `\__hook_label_pair:nn` {⟨*hook*⟩} {⟨l_A⟩} {⟨l_B⟩} to build a string $l_B|l_A$ with a fixed order, and use `\__hook_label_ordered:nnTF` to apply the correct rule to the pair of labels, depending if it was sorted or not.

```
\cs_new_protected:Npn \__hook_rule_before_gset:nnn #1#2#3
  {
    \tl_gset:cx { g__hook_#1_rule_ \__hook_label_pair:nn {#2} {#3} _tl }
      { \__hook_label_ordered:nnTF {#2} {#3} { < } { > } }
  }
\cs_new_eq:cN { __hook_rule_<_gset:nnn } \__hook_rule_before_gset:nnn

\cs_new_protected:Npn \__hook_rule_after_gset:nnn #1#2#3
  {
    \tl_gset:cx { g__hook_#1_rule_ \__hook_label_pair:nn {#3} {#2} _tl }
      { \__hook_label_ordered:nnTF {#3} {#2} { < } { > } }
  }
\cs_new_eq:cN { __hook_rule_>_gset:nnn } \__hook_rule_after_gset:nnn
```

(*End definition for* `\__hook_rule_before_gset:nnn` *and others.*)

`\__hook_rule_voids_gset:nnn`

This rule removes (clears, actually) the code from label `#3` if label `#2` is in the hook `#1`.

```
\cs_new_protected:Npn \__hook_rule_voids_gset:nnn #1#2#3
  {
    \tl_gset:cx { g__hook_#1_rule_ \__hook_label_pair:nn {#2} {#3} _tl }
      { \__hook_label_ordered:nnTF {#2} {#3} { -> } { <- } }
  }
```

(*End definition for* `\__hook_rule_voids_gset:nnn`.)

__hook_rule_incompatible-error_gset:nnn
__hook_rule_incompatible-warning_gset:nnn

These relations make an error/warning if labels #2 and #3 appear together in hook #1.

```
\cs_new_protected:cpn { __hook_rule_incompatible-error_gset:nnn } #1#2#3
  { \tl_gset:cn { g__hook_#1_rule_ \__hook_label_pair:nn {#2} {#3} _tl } { xE } }
\cs_new_protected:cpn { __hook_rule_incompatible-warning_gset:nnn } #1#2#3
  { \tl_gset:cn { g__hook_#1_rule_ \__hook_label_pair:nn {#2} {#3} _tl } { xW } }
```

(*End definition for* __hook_rule_incompatible-error_gset:nnn *and* __hook_rule_incompatible-warning_gset:nnn.)

__hook_rule_unrelated_gset:nnn
__hook_rule_gclear:nnn

Undo a setting. __hook_rule_unrelated_gset:nnn doesn't need to do anything, since we use __hook_rule_gclear:nnn before setting any rule.

```
\cs_new_protected:Npn \__hook_rule_unrelated_gset:nnn #1#2#3 { }
\cs_new_protected:Npn \__hook_rule_gclear:nnn #1#2#3
  { \cs_undefine:c { g__hook_#1_rule_ \__hook_label_pair:nn {#2} {#3} _tl } }
```

(*End definition for* __hook_rule_unrelated_gset:nnn *and* __hook_rule_gclear:nnn.)

__hook_label_pair:nn

Ensure that the lexically greater label comes first.

```
\cs_new:Npn \__hook_label_pair:nn #1#2
  {
    \if_case:w \__hook_str_compare:nn {#1} {#2} \exp_stop_f:
          #1 | #1 %  0
    \or:   #1 | #2 % +1
    \else: #2 | #1 % -1
    \fi:
  }
```

(*End definition for* __hook_label_pair:nn.)

__hook_label_ordered_p:nn
__hook_label_ordered:nn*TF*

Check that labels #1 and #2 are in the correct order (as returned by __hook_label_pair:nn) and if so return true, else return false.

```
\prg_new_conditional:Npnn \__hook_label_ordered:nn #1#2 { TF }
  {
    \if_int_compare:w \__hook_str_compare:nn {#1} {#2} > 0 \exp_stop_f:
      \prg_return_true:
    \else
      \prg_return_false:
    \fi:
  }
```

(*End definition for* __hook_label_ordered:nnTF.)

__hook_if_label_case:nnnnn

To avoid doing the string comparison twice in __hook_initialize_single:NNNn (once with \str_if_eq:nn and again with __hook_label_ordered:nn), we use a three-way branching macro that will compare #1 and #2 and expand to \use_i:nnn if they are equal, \use_ii:nn if #1 is lexically greater, and \use_iii:nn otherwise.

```
\cs_new:Npn \__hook_if_label_case:nnnnn #1#2
   {
     \cs:w use_
       \if_case:w \__hook_str_compare:nn {#1} {#2}
          i \or: ii \else: iii \fi: :nnn
     \cs_end:
   }
```

(*End definition for* __hook_if_label_case:nnnnn.)

__hook_update_hook_code:n

Before \begin{document} this does nothing, in the body it reinitializes the hook code using the altered data.

```
\cs_new_eq:NN \__hook_update_hook_code:n \use_none:n
```

(*End definition for* __hook_update_hook_code:n.)

__hook_initialize_all:

Initialize all known hooks (at \begin{document}), i.e., update the fast execution token lists to hold the necessary code in the right order.

```
\cs_new_protected:Npn \__hook_initialize_all: {
```

First we change __hook_update_hook_code:n which so far was a no-op to now initialize one hook. This way any later updates to the hook will run that code and also update the execution token list.

```
  \cs_gset_eq:NN \__hook_update_hook_code:n \__hook_initialize_hook_code:n
```

Now we loop over all hooks that have been defined and update each of them.

```
  \__hook_debug:n { \prop_gclear:N \g__hook_used_prop }
  \seq_map_inline:Nn \g__hook_all_seq
      {
        \__hook_update_hook_code:n {##1}
      }
```

If we are debugging we show results hook by hook for all hooks that have data.

```
  \__hook_debug:n
     { \iow_term:x{^^JAll~ initialized~ (non-empty)~ hooks:}
       \prop_map_inline:Nn \g__hook_used_prop
           { \iow_term:x{^^J~ ##1~ ->~
               \exp_not:v {__hook~##1}~ }
           }
     }
```

After all hooks are initialized we change the "use" to just call the hook code and not initialize it (as it was done in the preamble.

```
  \cs_gset_eq:NN \hook_use:n \__hook_use_initialized:n
  \cs_gset_eq:NN \__hook_preamble_hook:n \use_none:n
}
```

(*End definition for* __hook_initialize_all:.)

__hook_initialize_hook_code:n

Initializing or reinitializing the fast execution hook code. In the preamble this is selectively done in case a hook gets used and at \begin{document} this is done for all hooks and afterwards only if the hook code changes.

```
\cs_new_protected:Npn \__hook_initialize_hook_code:n #1
  {
    \debug_suspend:
    \__hook_debug:n{ \iow_term:x{^^JUpdate~ code~ for~ hook~
                                    '#1' \on@line :^^J} }
```

This does the sorting and the updates. First thing we do is to check if a legacy hook macro exists and if so we add it to the hook under the label legacy. This might make the hook non-empty so we have to do this before the then following test.

```
    \__hook_include_legacy_code_chunk:n {#1}
```

If there aren't any code chunks for the current hook, there is no point in even starting the sorting routine so we make a quick test for that and in that case just update `\__hook` ⟨*hook*⟩ to hold the next code. If there are code chunks we call `\__hook_initialize_single:NNNn` and pass to it ready made csnames as they are needed several times inside. This way we save a bit on processing time if we do that up front.

```
    \hook_if_exist:nT {#1}
      {
        \prop_if_empty:cTF {g__hook_#1_code_prop}
          { \tl_gset:co { __hook~#1 }
                        { \cs:w __hook_next~#1 \cs_end: } }
          {
```

By default the algorithm sorts the code chunks and then saves the result in a token list for fast execution by adding the code one after another using `\tl_gput_right:NV`. When we sort code for a reversed hook, all we have to do is to add the code chunks in the opposite order into the token list. So all we have to do in preparation is to change two definitions used later on.

```
            \__hook_if_reversed:nTF {#1}
              { \cs_set_eq:NN \__hook_tl_gput:NV    \tl_gput_left:NV
                \cs_set_eq:NN \__hook_clist_gput:NV \clist_gput_left:NV  }
              { \cs_set_eq:NN \__hook_tl_gput:NV    \tl_gput_right:NV
                \cs_set_eq:NN \__hook_clist_gput:NV \clist_gput_right:NV }
```

When sorting, some relations (namely `voids`) need to act destructively on the code property lists to remove code that shouldn't appear in the sorted hook token list, so we temporarily save the old code property list so that it can be restored later.

```
            \prop_set_eq:Nc \l__hook_work_prop { g__hook_#1_code_prop }
            \__hook_initialize_single:cccn
              { __hook~#1 } { __hook_next~#1 }
              { g__hook_#1_labels_clist } {#1}
```

For debug display we want to keep track of those hooks that actually got code added to them, so we record that in plist. We use a plist to ensure that we record each hook name only once, i.e., we are only interested in storing the keys and the value is arbitrary.

```
            \__hook_debug:n{ \exp_args:NNx \prop_gput:Nnn
                                           \g__hook_used_prop {#1}{} }
          }
      }
    \debug_resume:
  }
```

(*End definition for* `\__hook_initialize_hook_code:n`.)

`\__hook_tl_csname:n`
`\__hook_seq_csname:n`

It is faster to pass a single token and expand it when necessary than to pass a bunch of character tokens around.

FMi: note to myself: verify

```
\cs_new:Npn \__hook_tl_csname:n #1 { l__hook_label_#1_tl }
\cs_new:Npn \__hook_seq_csname:n #1 { l__hook_label_#1_seq }
```

(*End definition for* `\__hook_tl_csname:n` *and* `\__hook_seq_csname:n`.)

`\l__hook_labels_seq`
`\l__hook_labels_int`
`\l__hook_front_tl`
`\l__hook_rear_tl`
`\l__hook_label_0_tl`

For the sorting I am basically implementing Knuth's algorithm for topological sorting as given in TAOCP volume 1 pages 263–266. For this algorithm we need a number of local variables:

- List of labels used in the current hook to label code chunks:

```
\seq_new:N \l__hook_labels_seq
```

- Number of labels used in the current hook. In Knuth's algorithm this is called N:

```
\int_new:N \l__hook_labels_int
```

- The sorted code list to be build is managed using two pointers one to the front of the queue and one to the rear. We model this using token list pointers. Knuth calls them F and R:

```
\tl_new:N \l__hook_front_tl
\tl_new:N \l__hook_rear_tl
```

- The data for the start of the queue is kept in this token list, it corresponds to what Don calls `QLINK[0]` but since we aren't manipulating individual words in memory it is slightly differently done:

```
\tl_new:c { \__hook_tl_csname:n { 0 } }
```

(*End definition for* `\l__hook_labels_seq` *and others.*)

__hook_initialize_single:NNNn
__hook_initialize_single:cccn

`\__hook_initialize_single:NNNn` implements the sorting of the code chunks for a hook and saves the result in the token list for fast execution (`#3`). The arguments are ⟨*hook-code-plist*⟩, ⟨*hook-code-tl*⟩, ⟨*hook-next-code-tl*⟩, ⟨*hook-ordered-labels-clist*⟩ and ⟨*hook-name*⟩ (the latter is only used for debugging—the ⟨*hook-rule-plist*⟩ is accessed using the ⟨*hook-name*⟩).

The additional complexity compared to Don's algorithm is that we do not use simple positive integers but have arbitrary alphanumeric labels. As usual Don's data structures are chosen in a way that one can omit a lot of tests and I have mimicked that as far as possible. The result is a restriction I do not test for at the moment: a label can't be equal to the number 0!

FMi: Needs checking for, just in case

```
\cs_new_protected:Npn \__hook_initialize_single:NNNn #1#2#3#4
  {
```

Step T1: Initialize the data structure ...

```
    \seq_clear:N \l__hook_labels_seq
    \int_zero:N  \l__hook_labels_int
```

Store the name of the hook:

```
    \tl_set:Nn \l__hook_cur_hook_tl {#4}
```

We loop over the property list holding the code and record all labels listed there. Only rules for those labels are of interest to us. While we are at it we count them (which gives us the N in Knuth's algorithm. The prefix `label_` is added to the variables to ensure that labels named `front`, `rear`, `labels`, or `return` don't interact with our code.

```
    \prop_map_inline:Nn \l__hook_work_prop
       {
         \int_incr:N \l__hook_labels_int
         \seq_put_right:Nn \l__hook_labels_seq {##1}
         \tl_set:cn { \__hook_tl_csname:n {##1} }{0}
         \seq_clear_new:c { \__hook_seq_csname:n {##1} }
       }
```

Steps T2 and T3: Sort the relevant rules into the data structure...

This loop constitutes a square matrix of the labels in `\l__hook_work_prop` in the vertical and the horizontal directions. However since the rule $l_A\langle rel\rangle l_B$ is the same as $l_B\langle rel\rangle^{-1} l_A$ we can cut the loop short at the diagonal of the matrix (*i.e.*, when both labels are equal), saving a good amount of time. The way the rules were set up (see the implementation of `\__hook_rule_before_gset:nnn` above) ensures that we have no rule in the ignored side of the matrix, and all rules are seen. The rules are applied in `\__hook_apply_label_pair:nnn`, which takes the properly-ordered pair of labels as argument.

```
\prop_map_inline:Nn \l__hook_work_prop
  {
    \prop_map_inline:Nn \l__hook_work_prop
      {
        \__hook_if_label_case:nnnnn {##1} {####1}
          { \prop_map_break: }
          { \__hook_apply_label_pair:nnn {##1} {####1} }
          { \__hook_apply_label_pair:nnn {####1} {##1} }
              {#4}
      }
  }
```

Take a breath and take a look at the data structures that have been set up:

```
\__hook_debug:n { \__hook_debug_label_data:N \l__hook_work_prop }
```

Step T4:

```
\tl_set:Nn \l__hook_rear_tl { 0 }
\tl_set:cn { \__hook_tl_csname:n { 0 } } { 0 }
\seq_map_inline:Nn \l__hook_labels_seq
  {
    \int_compare:nNnT { \cs:w \__hook_tl_csname:n {##1} \cs_end: } = 0
        {
          \tl_set:cn { \__hook_tl_csname:n { \l__hook_rear_tl } }{##1}
          \tl_set:Nn \l__hook_rear_tl {##1}
        }
  }
\tl_set_eq:Nc \l__hook_front_tl { \__hook_tl_csname:n { 0 } }

\tl_gclear:N #1
\clist_gclear:N #3
```

The whole loop combines steps T5–T7:

```
\bool_while_do:nn { ! \str_if_eq_p:Vn \l__hook_front_tl { 0 } }
  {
```

This part is step T5:

```
    \int_decr:N \l__hook_labels_int
    \prop_get:NVN \l__hook_work_prop \l__hook_front_tl \l__hook_return_tl
    \__hook_tl_gput:NV #1 \l__hook_return_tl

    \__hook_clist_gput:NV #3 \l__hook_front_tl
    \__hook_debug:n{ \iow_term:x{Handled~ code~ for~ \l__hook_front_tl} }
```

This is step T6 except that we don't use a pointer P to move through the successors, but instead use `##1` of the mapping function.

```
    \seq_map_inline:cn { \__hook_seq_csname:n { \l__hook_front_tl } }
      {
```

```
            \tl_set:cx { \__hook_tl_csname:n {##1} }
                       { \int_eval:n
                           { \cs:w \__hook_tl_csname:n {##1} \cs_end: - 1 }
                       }
            \int_compare:nNnT
                { \cs:w \__hook_tl_csname:n {##1} \cs_end: } = 0
                {
                  \tl_set:cn { \__hook_tl_csname:n { \l__hook_rear_tl } } {##1}
                  \tl_set:Nn \l__hook_rear_tl            {##1}
                }
          }
```

and step T7:

```
        \tl_set_eq:Nc \l__hook_front_tl
                      { \__hook_tl_csname:n { \l__hook_front_tl } }
```

This is step T8: If we haven't moved the code for all labels (i.e., if `\l__hook_labels_int` is still greater than zero) we have a loop and our partial order can't be flattened out.

```
      }
    \int_compare:nNnF \l__hook_labels_int = 0
      {
        \iow_term:x{====================}
        \iow_term:x{Error:~ label~ rules~ are~ incompatible:}
```

This is not really the information one needs in the error case but will do for now ...

FMi: fix

```
        \__hook_debug_label_data:N \l__hook_work_prop
        \iow_term:x{====================}
      }
```

After we have added all hook code to `#1` we finish it off with adding extra code for a one time execution. That is stored in `#2` but is normally empty.

```
    \tl_gput_right:Nn #1 {#2}
  }
\cs_generate_variant:Nn \__hook_initialize_single:NNNn {ccc}
```

(*End definition for* `\__hook_initialize_single:NNNn`.)

`\__hook_tl_gput:NV`
`\__hook_clist_gput:NV`

These append either on the right (normal hook) or on the left (reversed hook). This is setup up in `\__hook_initialize_hook_code:n`, elsewhere their behavior is undefined.

```
\cs_new:Npn \__hook_tl_gput:NV     {\ERROR}
\cs_new:Npn \__hook_clist_gput:NV  {\ERROR}
```

(*End definition for* `\__hook_tl_gput:NV` *and* `\__hook_clist_gput:NV`.)

`\__hook_apply_label_pair:nnn`
`\__hook_label_if_exist_apply:nnnF`

This is the payload of steps T2 and T3 executed in the loop described above. This macro assumes `#1` and `#2` are ordered, which means that any rule pertaining the pair `#1` and `#2` is `\g__hook_`⟨*hook*⟩`_rule_#1|#2_tl`, and not `\g__hook_`⟨*hook*⟩`_rule_#2|#1_tl`. This also saves a great deal of time since we only need to check the order of the labels once.

The arguments here are ⟨*label1*⟩, ⟨*label2*⟩, ⟨*hook*⟩, and ⟨*hook-code-plist*⟩. We are about to apply the next rule and enter it into the data structure. `\__hook_apply_label_pair:nnn` will just call `\__hook_label_if_exist_apply:nnnF` for the ⟨*hook*⟩, and if no rule is found, also try the ⟨*hook*⟩ name `??` denoting a default hook rule.

__hook_label_if_exist_apply:nnnF will check if the rule exists for the given hook, and if so call __hook_apply_rule:nnn.

```
\cs_new_protected:Npn \__hook_apply_label_pair:nnn #1#2#3
  {
```

Extra complication: as we use default rules and local hook specific rules we first have to check if there is a local rule and if that exist use it. Otherwise check if there is a default rule and use that.

```
    \__hook_label_if_exist_apply:nnnF {#1} {#2} {#3}
      {
```

If there is no hook-specific rule we check for a default one and use that if it exists.

```
        \__hook_label_if_exist_apply:nnnF {#1} {#2} { ?? } { }
      }
  }
\cs_new_protected:Npn \__hook_label_if_exist_apply:nnnF #1#2#3
  {
    \if_cs_exist:w g__hook_ #3 _rule_ #1 | #2 _tl \cs_end:
```

What to do precisely depends on the type of rule we have encountered. If it is a `before` rule it will be handled by the algorithm but other types need to be managed differently. All this is done in __hook_apply_rule:nnnN.

```
      \__hook_apply_rule:nnn {#1} {#2} {#3}
      \exp_after:wN \use_none:n
    \else:
      \use:nn
    \fi:
  }
```

(*End definition for* __hook_apply_label_pair:nnn *and* __hook_label_if_exist_apply:nnnF.)

__hook_apply_rule:nnn

This is the code executed in steps T2 and T3 while looping through the matrix This is part of step T3. We are about to apply the next rule and enter it into the data structure. The arguments are ⟨*label1*⟩, ⟨*label2*⟩, ⟨*hook-name*⟩, and ⟨*hook-code-plist*⟩.

```
\cs_new_protected:Npn \__hook_apply_rule:nnn #1#2#3
  {
    \cs:w __hook_apply_
      \cs:w g__hook_#3_reversed_tl \cs_end: rule_
        \cs:w g__hook_ #3 _rule_ #1 | #2 _tl \cs_end: :nnn \cs_end:
      {#1} {#2} {#3}
  }
```

(*End definition for* __hook_apply_rule:nnn.)

__hook_apply_rule_<:nnn
__hook_apply_rule_>:nnn

The most common cases are < and > so we handle that first. They are relations $\prec$ and $\succ$ in TAOCP, and they dictate sorting.

```
\cs_new_protected:cpn { __hook_apply_rule_<:nnn } #1#2#3
  {
    \__hook_debug:n { \__hook_msg_pair_found:nnn {#1} {#2} {#3} }
    \tl_set:cx { \__hook_tl_csname:n {#2} }
       { \int_eval:n{ \cs:w \__hook_tl_csname:n {#2} \cs_end: + 1 } }
    \seq_put_right:cn{ \__hook_seq_csname:n {#1} }{#2}
  }
\cs_new_protected:cpn { __hook_apply_rule_>:nnn } #1#2#3
  {
```

```
    \__hook_debug:n { \__hook_msg_pair_found:nnn {#1} {#2} {#3} }
    \tl_set:cx { \__hook_tl_csname:n {#1} }
       { \int_eval:n{ \cs:w \__hook_tl_csname:n {#1} \cs_end: + 1 } }
    \seq_put_right:cn{ \__hook_seq_csname:n {#2} }{#1}
  }
```

(*End definition for* `\__hook_apply_rule_<:nnn` *and* `\__hook_apply_rule_>:nnn`.)

`\__hook_apply_rule_xE:nnn`
`\__hook_apply_rule_xW:nnn`

These relations make two labels incompatible within a hook. `xE` makes raises an error if the labels are found in the same hook, and `xW` makes it a warning.

```
\cs_new_protected:cpn { __hook_apply_rule_xE:nnn } #1#2#3
  {
    \__hook_debug:n { \__hook_msg_pair_found:nnn {#1} {#2} {#3} }
    \msg_error:nnnnnn { hooks } { labels-incompatible }
      {#1} {#2} {#3} { 1 }
    \use:c { __hook_apply_rule_->:nnn } {#1} {#2} {#3}
    \use:c { __hook_apply_rule_<-:nnn } {#1} {#2} {#3}
  }
\cs_new_protected:cpn { __hook_apply_rule_xW:nnn } #1#2#3
  {
    \__hook_debug:n { \__hook_msg_pair_found:nnn {#1} {#2} {#3} }
    \msg_warning:nnnnnn { hooks } { labels-incompatible }
      {#1} {#2} {#3} { 0 }
  }
```

(*End definition for* `\__hook_apply_rule_xE:nnn` *and* `\__hook_apply_rule_xW:nnn`.)

`\__hook_apply_rule_->:nnn`
`\__hook_apply_rule_<-:nnn`

If we see `->` we have to drop code for label `#3` and carry on. We could do a little better and drop everything for that label since it doesn't matter where we sort in the empty code. However that would complicate the algorithm a lot with little gain.[8] So we still unnecessarily try to sort it in and depending on the rules that might result in a loop that is otherwise resolved. If that turns out to be a real issue, we can improve the code.

Here the code is removed from `\l__hook_cur_hook_tl` rather than `#3` because the latter may be `??`, and the default hook doesn't store any code. Removing from `\l__hook_cur_hook_tl` makes default rules `->` and `<-` work properly.

```
\cs_new_protected:cpn { __hook_apply_rule_->:nnn } #1#2#3
  {
    \__hook_debug:n
       {
         \__hook_msg_pair_found:nnn {#1} {#2} {#3}
         \iow_term:x{--->~ Drop~ '#2'~ code~ from~
           \iow_char:N \\ g__hook_ \l__hook_cur_hook_tl _code_prop ~
           because~ of~ '#1' }
       }
    \prop_put:Nnn \l__hook_work_prop {#2} { }
  }
\cs_new_protected:cpn { __hook_apply_rule_<-:nnn } #1#2#3
  {
    \__hook_debug:n
       {
         \__hook_msg_pair_found:nnn {#1} {#2} {#3}
```

[8] This also hase the advantage that the result of the sorting doesn't change which might otherwise (for unrelated chunks) if we aren't careful.

```
        \iow_term:x{--->~ Drop~ '#1'~ code~ from~
          \iow_char:N \\ g__hook_ \l__hook_cur_hook_tl _code_prop ~
          because~ of~ '#2' }
      }
    \prop_put:Nnn \l__hook_work_prop {#1} { }
  }
```

(*End definition for* `\__hook_apply_rule_->:nnn` *and* `\__hook_apply_rule_<-:nnn`.)

`\__hook_apply_-rule_<:nnn`
`\__hook_apply_-rule_>:nnn`
`\__hook_apply_-rule_<-:nnn`
`\__hook_apply_-rule_->:nnn`
`\__hook_apply_-rule_x:nnn`

Reversed rules.

```
\cs_new_eq:cc { __hook_apply_-rule_<:nnn  } { __hook_apply_rule_>:nnn }
\cs_new_eq:cc { __hook_apply_-rule_>:nnn  } { __hook_apply_rule_<:nnn }
\cs_new_eq:cc { __hook_apply_-rule_<-:nnn } { __hook_apply_rule_<-:nnn }
\cs_new_eq:cc { __hook_apply_-rule_->:nnn } { __hook_apply_rule_->:nnn }
\cs_new_eq:cc { __hook_apply_-rule_xE:nnn  } { __hook_apply_rule_xE:nnn }
\cs_new_eq:cc { __hook_apply_-rule_xW:nnn  } { __hook_apply_rule_xW:nnn }
```

(*End definition for* `\__hook_apply_-rule_<:nnn` *and others.*)

`\__hook_msg_pair_found:nnn`

A macro to avoid moving this many tokens around.

```
\cs_new_protected:Npn \__hook_msg_pair_found:nnn #1#2#3
  {
    \iow_term:x{~ \str_if_eq:nnTF {#3} {??} {default} {~normal} ~
        rule~ \__hook_label_pair:nn {#1} {#2}:~
        \use:c { g__hook_#3_rule_ \__hook_label_pair:nn {#1} {#2} _tl } ~
        found}
  }
```

(*End definition for* `\__hook_msg_pair_found:nnn`.)

`\__hook_debug_label_data:N`

```
\cs_new_protected:Npn \__hook_debug_label_data:N #1 {
  \iow_term:x{Code~ labels~ for~ sorting:}
  \iow_term:x{~ \seq_use:Nnnn\l__hook_labels_seq {~and~}{,~}{~and~} }
  \iow_term:x{^^J Data~ structure~ for~ label~ rules:}
  \prop_map_inline:Nn #1
      {
        \iow_term:x{~ ##1~ =~ \tl_use:c{ \__hook_tl_csname:n {##1} }~ ->~
          \seq_use:cnnn{ \__hook_seq_csname:n {##1} }{~->~}{~->~}{~->~}
        }
      }
  \iow_term:x{}
}
```

(*End definition for* `\__hook_debug_label_data:N`.)

`\hook_show:n`
`\hook_log:n`
`\__hook_log_line:x`
`\__hook_log_line_indent:x`
`\__hook_log:nN`

This writes out information about the hook given in its argument onto the `.log` file and the terminal, if `\show_hook:n` is used. Internally both share the same structure, except that at the end, `\hook_show:n` triggers TEX's prompt.

```
\cs_new_protected:Npn \hook_log:n #1
  {
    \cs_set_eq:NN \__hook_log_cmd:x \iow_log:x
    \__hook_normalize_hook_args:Nn \__hook_log:nN {#1} \tl_log:x
  }
\cs_new_protected:Npn \hook_show:n #1
```

```
  {
    \cs_set_eq:NN \__hook_log_cmd:x \iow_term:x
    \__hook_normalize_hook_args:Nn \__hook_log:nN {#1} \tl_show:x
  }
\cs_new_protected:Npn \__hook_log_line:x #1
  { \__hook_log_cmd:x { >~#1 } }
\cs_new_protected:Npn \__hook_log_line_indent:x #1
  { \__hook_log_cmd:x { >~\@spaces #1 } }
\cs_new_protected:Npn \__hook_log:nN #1 #2
  {
    \__hook_preamble_hook:n {#1}
    \__hook_log_cmd:x { ^^J ->~The~hook~'#1': }
    \hook_if_exist:nF {#1}
      { \__hook_log_line:x { is~not~declared! } }
    \__hook_if_exist:nTF {#1}
      {
        \__hook_log_line:x { Code~chunks: }
        \prop_if_empty:cTF { g__hook_#1_code_prop }
          { \__hook_log_line_indent:x { --- } }
          {
            \prop_map_inline:cn { g__hook_#1_code_prop }
              { \__hook_log_line_indent:x { ##1~->~\tl_to_str:n {##2} } }
          }
        \__hook_log_line:x { Extra~code~for~next~invocation:}
        \__hook_log_line_indent:x
          {
            \tl_if_empty:cTF { __hook_next~#1 }
              { --- }
```

If the token list is not empty we want to display it but without the first tokens (the code to clear itself) so we call a helper command to get rid of them.

```
              { ->~ \exp_args:Nv \__hook_log_next_code:n { __hook_next~#1 } }
          }
```

Loop through the rules in a hook and for every rule found, print it. If no rule is there, print `---`. The boolean `\l__hook_tmpa_bool` here indicates if the hook has no rules.

```
        \__hook_log_line:x { Rules: }
        \bool_set_true:N \l__hook_tmpa_bool
        \__hook_list_rules:nn {#1}
          {
            \bool_set_false:N \l__hook_tmpa_bool
            \__hook_log_line_indent:x
              {
                ##2~ with~
                \str_if_eq:nnT {##3} {??} { default~ }
                relation~ ##1
              }
          }
        \bool_if:NT \l__hook_tmpa_bool
          { \__hook_log_line_indent:x { --- } }
```

When the hook is declared (that is, the sorting algorithm is applied to that hook) and not empty

```
        \bool_lazy_and:nnTF
            { \hook_if_exist_p:n {#1} }
            { ! \hook_if_empty_p:n {#1} }
          {
            \__hook_log_line:x
              {
                Execution~order
                \bool_if:NTF \l__hook_tmpa_bool
                  { \__hook_if_reversed:nT {#1} { ~(after~reversal) } }
                  { ~(after~
                    \__hook_if_reversed:nT {#1} { reversal~and~ }
                    applying~rules)
                  } :
              }
            #2 % \tl_show:n
              {
                \@spaces
                \clist_if_empty:cTF { g__hook_#1_labels_clist }
                  { --- }
                  { \clist_use:cn {g__hook_#1_labels_clist} { ,~ } }
              }
          }
          {
            #2
              {
                Hook~ \hook_if_exist:nTF {#1}
                  {code~pool~empty} {not~declared}
              }
          }
      }
      { #2 { The~hook~is~empty. } }
  }
```

To display the code for next invocation only (i.e., from `\AddToHookNext` we have to remove the first two tokens at the front which are `\tl_gclear:N` and the token list to clear.

```
\cs_new:Npn \__hook_log_next_code:n #1
  { \exp_args:No \tl_to_str:n { \use_none:nn #1 } }
```

`\__hook_log_next_code:n`

(*End definition for* `\hook_show:n` *and others. These functions are documented on page* 111.)

`\__hook_list_rules:nn`
`\__hook_list_one_rule:nnn`
`\__hook_list_if_rule_exists:nnnF`

This macro takes a ⟨*hook*⟩ and an ⟨*inline function*⟩ and loops through each pair of ⟨*labels*⟩ in the ⟨*hook*⟩, and if there is a relation between this pair of ⟨*labels*⟩, the ⟨*inline function*⟩ is executed with `#1` = ⟨*relation*⟩, `#2` = ⟨$label_1$⟩ | ⟨$label_2$⟩, and `#3` = ⟨*hook*⟩ (the latter may be the argument `#1` to `\__hook_list_rules:nn`, or `??` if it is a default rule).

```
\cs_new_protected:Npn \__hook_list_rules:nn #1 #2
  {
    \cs_set_protected:Npn \__hook_tmp:w ##1 ##2 ##3 {#2}
    \prop_map_inline:cn { g__hook_#1_code_prop }
      {
        \prop_map_inline:cn { g__hook_#1_code_prop }
          {
            \__hook_if_label_case:nnnnn {##1} {####1}
              { \prop_map_break: }
```

```
              { \__hook_list_one_rule:nnn {##1} {####1} }
              { \__hook_list_one_rule:nnn {####1} {##1} }
                  {#1}
          }
      }
  }
```

These two are quite similar to `\__hook_apply_label_pair:nnn` and `\__hook_label_if_exist_apply:nnnF`, respectively, but rather than applying the rule, they pass it to the ⟨*inline function*⟩.

```
\cs_new_protected:Npn \__hook_list_one_rule:nnn #1#2#3
  {
    \__hook_list_if_rule_exists:nnnF {#1} {#2} {#3} { }
      { \__hook_list_if_rule_exists:nnnF {#1} {#2} { ?? } { } }
  }
\cs_new_protected:Npn \__hook_list_if_rule_exists:nnnF #1#2#3
  {
    \if_cs_exist:w g__hook_ #3 _rule_ #1 | #2 _tl \cs_end:
      \exp_args:Nv \__hook_tmp:w
        { g__hook_ #3 _rule_ #1 | #2 _tl } { #1 | #2 } {#3}
    \fi:
  }
```

(*End definition for* `\__hook_list_rules:nn`, `\__hook_list_one_rule:nnn`, *and* `\__hook_list_if_rule_exists:nnnF`.)

`\__hook_debug_print_rules:n`

A shorthand for debugging that prints similar to `\prop_show:N`.

```
\cs_new_protected:Npn \__hook_debug_print_rules:n #1
  {
    \iow_term:n { The~hook~#1~contains~the~rules: }
    \cs_set_protected:Npn \__hook_tmp:w ##1
      {
        \__hook_list_rules:nn {#1}
          {
            \iow_term:x
              {
                > ##1 {####2} ##1 => ##1 {####1}
                \str_if_eq:nnT {####3} {??} { ~(default) }
              }
          }
      }
    \exp_args:No \__hook_tmp:w { \use:nn { ~ } { ~ } }
  }
```

(*End definition for* `\__hook_debug_print_rules:n`.)

3.8 Specifying code for next invocation

`\hook_gput_next_code:nn`
`%␣␣␣␣\__hook_gput_next_code:nn`
`\__hook_gput_next_do:nn`
`\__hook_gput_next_do:Nnn`
`\__hook_clear_next:n`

```
\cs_new_protected:Npn \hook_gput_next_code:nn #1
  { \__hook_normalize_hook_args:Nn \__hook_gput_next_code:nn {#1} }
\cs_new_protected:Npn \__hook_gput_next_code:nn #1 #2
  {
    \debug_suspend:
```

```
    \__hook_declare:n {#1}
    \hook_if_exist:nTF {#1}
      { \__hook_gput_next_do:nn {#1} {#2} }
      { \__hook_try_declaring_generic_next_hook:nn {#1} {#2} }
    \debug_resume:
  }
\cs_new_protected:Npn \__hook_gput_next_do:nn #1
  {
    \exp_args:Nc \__hook_gput_next_do:Nnn
      { __hook_next~#1 } {#1}
  }
```

First check if the "next code" token list is empty: if so we need to add a `\tl_gclear:c` to clear it, so the code lasts for one usage only. The token list is cleared early so that nested usages don't get lost. `\tl_gclear:c` is used instead of `\tl_gclear:N` in case the hook is used in an expansion-only context, so the token list doesn't expand before `\tl_gclear:N`: that would make an infinite loop. Also in case the main code token list is empty, the hook code has to be updated to add the next execution token list.

```
\cs_new_protected:Npn \__hook_gput_next_do:Nnn #1 #2
  {
    \tl_if_empty:cT { __hook~#2 }
      { \__hook_update_hook_code:n {#2} }
    \tl_if_empty:NT #1
      { \tl_gset:Nn #1 { \__hook_clear_next:n {#2} } }
    \tl_gput_right:Nn #1
  }
\cs_new_protected:Npn \__hook_clear_next:n #1
  { \cs_gset_eq:cN { __hook_next~#1 } \c_empty_tl }
```

(*End definition for* `\hook_gput_next_code:nn` *and others. This function is documented on page* 110.)

3.9 Using the hook

`\hook_use:n`
`\__hook_use_initialized:n`
`\__hook_use_undefined:w`
`\__hook_use_end:`
`\__hook_preamble_hook:n`

`\hook_use:n` as defined here is used in the preamble, where hooks aren't initialized by default. `\__hook_use_initialized:n` is also defined, which is the non-`\protected` version for use within the document. Their definition is identical, except for the `\__hook_preamble_hook:n` (which wouldn't hurt in the expandable version, but it would be an unnecessary extra expansion).

`\__hook_use_initialized:n` holds the expandable definition while in the preamble. `\__hook_preamble_hook:n` initializes the hook in the preamble, and is redefined to `\use_none:n` at `\begin{document}`.

Both versions do the same internally: check if the hook exist as given, and if so use it as quickly as possible. If it doesn't exist, the a call to `\__hook_use:wn` checks for file hooks.

At `\begin{document}`, all hooks are initialized, and any change in them causes an update, so `\hook_use:n` can be made expandable. This one is better not protected so that it can expand into nothing if containing no code. Also important in case of generic hooks that we do not generate a `\relax` as a side effect of checking for a csname. In contrast to the TEX low-level `\csname ...\endcsname` construct `\tl_if_exist:c` is careful to avoid this.

```
\cs_new_protected:Npn \hook_use:n #1
  {
```

```
    \tl_if_exist:cTF { __hook~#1 }
      {
        \__hook_preamble_hook:n {#1}
        \cs:w __hook~#1 \cs_end:
      }
      { \__hook_use:wn #1 / \s__hook_mark {#1} }
  }
\cs_new:Npn \__hook_use_initialized:n #1
  {
    \if_cs_exist:w __hook~#1 \cs_end:
    \else:
      \__hook_use_undefined:w
    \fi:
    \cs:w __hook~#1 \__hook_use_end:
  }
\cs_new:Npn \__hook_use_undefined:w #1 #2 __hook~#3 \__hook_use_end:
  {
    #1 % fi
    \__hook_use:wn #3 / \s__hook_mark {#3}
  }
\cs_new_protected:Npn \__hook_preamble_hook:n #1
  { \__hook_initialize_hook_code:n {#1} }
\cs_new_eq:NN \__hook_use_end: \cs_end:
```

(*End definition for* `\hook_use:n` *and others. This function is documented on page* 109.)

`\__hook_use:wn`
`\__hook_try_file_hook:n`
`\__hook_if_exist_use:n`

`\__hook_use:wn` does a quick check to test if the current hook is a file hook: those need a special treatment. If it is not, the hook does not exist. If it is, then `\__hook_try_file_hook:n` is called, and checks that the current hook is a file-specific hook using `\__hook_if_file_hook:wTF`. If it's not, then it's a generic `file/` hook and is used if it exist.

If it is a file-specific hook, it passes through the same normalization as during declaration, and then it is used if defined. `\__hook_if_exist_use:n` checks if the hook exist, and calls `\__hook_preamble_hook:n` if so, then uses the hook.

```
\cs_new:Npn \__hook_use:wn #1 / #2 \s__hook_mark #3
  {
    \str_if_eq:nnTF {#1} { file }
      { \__hook_try_file_hook:n {#3} }
      { } % Hook doesn't exist
  }
\cs_new_protected:Npn \__hook_try_file_hook:n #1
  {
    \__hook_if_file_hook:wTF #1 / / \s__hook_mark
      {
        \exp_args:Ne \__hook_if_exist_use:n
          { \exp_args:Ne \__hook_file_hook_normalize:n {#1} }
      }
      { \__hook_if_exist_use:n {#1} } % file/ generic hook (e.g. file/before)
  }
\cs_new_protected:Npn \__hook_if_exist_use:n #1
  {
    \tl_if_exist:cT { __hook~#1 }
      {
        \__hook_preamble_hook:n {#1}
```

```
        \cs:w __hook~#1 \cs_end:
      }
  }
```

(*End definition for* `\__hook_use:wn`, `\__hook_try_file_hook:n`, *and* `\__hook_if_exist_use:n`.)

`\hook_use_once:n` For hooks that can and should be used only once we have a special use command that remembers the hook name in `\g__hook_execute_immediately_prop`. This has the effect that any further code added to the hook is executed immediately rather than stored in the hook.

The code needs some gymnastics to prevent space trimming from the hook name, since `\hook_use:n` and `\hook_use_once:n` are documented to not trim spaces.

PhO: Should this raise an error if the hook doesn't exist?

```
\cs_new_protected:Npn \hook_use_once:n #1
  {
    \tl_if_exist:cT { __hook~#1 }
      {
        \tl_set:Nn \l__hook_return_tl {#1}
        \__hook_normalize_hook_args:Nn \__hook_use_once_store:n
          { \l__hook_return_tl }
        \hook_use:n {#1}
      }
  }
\cs_new_protected:Npn \__hook_use_once_store:n #1
  { \prop_gput:Nnn \g__hook_execute_immediately_prop {#1} { } }
```

(*End definition for* `\hook_use_once:n`. *This function is documented on page* 109.)

3.10 Querying a hook

Simpler data types, like token lists, have three possible states; they can exist and be empty, exist and be non-empty, and they may not exist, in which case emptiness doesn't apply (though `\tl_if_empty:N` returns false in this case).

Hooks are a bit more complicated: they have four possible states. A hook may exist or not, and either way it may or may not be empty (even a hook that doesn't exist may be non-empty).

A hook is said to be empty when no code was added to it, either to its permanent code pool, or to its "next" token list. The hook doesn't need to be declared to have code added to its code pool (it may happen that a package A defines a hook `foo`, but it's loaded after package B, which adds some code to that hook. In this case it is important that the code added by package B is remembered until package A is loaded).

A hook is said to exist when it was declared with `\hook_new:n` or some variant thereof.

`\hook_if_empty_p:n` `\hook_if_empty:nTF` Test if a hook is empty (that is, no code was added to that hook). A hook being empty means that *both* its `\g__hook_`⟨*hook*⟩`_code_prop` and its `\__hook_next` ⟨*hook*⟩ are empty.

```
\prg_new_conditional:Npnn \hook_if_empty:n #1 { p , T , F , TF }
  {
    \__hook_if_exist:nTF {#1}
      {
```

```
        \bool_lazy_and:nnTF
            { \prop_if_empty_p:c { g__hook_#1_code_prop } }
            { \tl_if_empty_p:c { __hook_next~#1 } }
          { \prg_return_true: }
          { \prg_return_false: }
      }
      { \prg_return_true: }
  }
```

(*End definition for* `\hook_if_empty:nTF`. *This function is documented on page* 110.)

`\hook_if_exist_p:n`
`\hook_if_exist:nTF`

A canonical way to test if a hook exists. A hook exists if the token list that stores the sorted code for that hook, `\__hook` ⟨*hook*⟩, exists. The property list `\g__hook_`⟨*hook*⟩`_code_prop` cannot be used here because often it is necessary to add code to a hook without knowing if such hook was already declared, or even if it will ever be (for example, in case the package that defines it isn't loaded).

```
\prg_new_conditional:Npnn \hook_if_exist:n #1 { p , T , F , TF }
  {
    \tl_if_exist:cTF { __hook~#1 }
      { \prg_return_true: }
      { \prg_return_false: }
  }
```

(*End definition for* `\hook_if_exist:nTF`. *This function is documented on page* 110.)

`\__hook_if_exist_p:n`
`\__hook_if_exist:nTF`

An internal check if the hook has already been declared with `\__hook_declare:n`. This means that the hook was already used somehow (a code chunk or rule was added to it), but it still wasn't declared with `\hook_new:n`.

```
\prg_new_conditional:Npnn \__hook_if_exist:n #1 { p , T , F , TF }
  {
    \prop_if_exist:cTF { g__hook_#1_code_prop }
      { \prg_return_true: }
      { \prg_return_false: }
  }
```

(*End definition for* `\__hook_if_exist:nTF`.)

`\__hook_if_reversed_p:n`
`\__hook_if_reversed:nTF`

An internal conditional that checks if a hook is reversed.

```
\prg_new_conditional:Npnn \__hook_if_reversed:n #1 { p , T , F , TF }
  {
    \if_int_compare:w \cs:w g__hook_#1_reversed_tl \cs_end: 1 < 0 \exp_stop_f:
      \prg_return_true:
    \else:
      \prg_return_false:
    \fi:
  }
```

(*End definition for* `\__hook_if_reversed:nTF`.)

3.11 Messages

```
\msg_new:nnnn { hooks } { labels-incompatible }
  {
    Labels~‘#1’~and~‘#2’~are~incompatible
    \str_if_eq:nnF {#3} {??} { ~in~hook~‘#3’ } .~
    \int_compare:nNnTF {#4} = { 1 }
      { The~ code~ for~ both~ labels~ will~ be~ dropped. }
      { You~ may~ see~ errors~ later. }
  }
  { LaTeX~found~two~incompatible~labels~in~the~same~hook.~
    This~indicates~an~incompatibility~between~packages.  }
```

```
\msg_new:nnnn { hooks } { exists }
    { Hook~‘#1’~ has~ already~ been~ declared. }
    { There~ already~ exists~ a~ hook~ declaration~ with~ this~
      name.\\
      Please~ use~ a~ different~ name~ for~ your~ hook.}
```

```
\msg_new:nnn { hooks } { empty-label }
  { Empty~code~label~\msg_line_context:.~Using~‘#1’~instead. }
```

```
\msg_new:nnnn { hooks } { unknown-rule }
  { Unknown~ relationship~ ‘#3’~
    between~ labels~ ‘#2’~ and~ ‘#4’~
    \str_if_eq:nnF {#1} {??} { ~in~hook~‘#1’ }. ~
    Perhaps~ a~ missspelling?
  }
  {
    The~ relation~ used~ not~ known~ to~ the~ system.~ Allowed~ values~ are~
    ‘before’~ or~ ‘<’,~
    ‘after’~ or~ ‘>’,~
    ‘incompatible-warning’,~
    ‘incompatible-error’,~
    ‘voids’~ or~
    ‘unrelated’.
  }
```

```
\msg_new:nnn { hooks } { should-not-happen }
  {
    ERROR!~This~should~not~happen.~#1 \\
    Please~report~at~https://github.com/latex3/latex2e.
  }
```

3.12 LaTeX 2_ε package interface commands

\NewHook
\NewReversedHook
\NewMirroredHookPair

Declaring new hooks ...

```
\NewDocumentCommand \NewHook             { m }{ \hook_new:n {#1} }
\NewDocumentCommand \NewReversedHook     { m }{ \hook_new_reversed:n {#1} }
\NewDocumentCommand \NewMirroredHookPair { mm }{ \hook_new_pair:nn {#1}{#2} }
```

(*End definition for* `\NewHook`, `\NewReversedHook`, *and* `\NewMirroredHookPair`. *These functions are documented on page* 100.)

\AddToHook

```
\NewDocumentCommand \AddToHook { m o +m }
  { \hook_gput_code:nnn {#1} {#2} {#3} }
```

(*End definition for* `\AddToHook`. *This function is documented on page* 102.)

`\AddToHookNext`

```
\NewDocumentCommand \AddToHookNext { m +m }
  { \hook_gput_next_code:nn {#1} {#2} }
```

(*End definition for* `\AddToHookNext`. *This function is documented on page* 103.)

`\RemoveFromHook`

```
\NewDocumentCommand \RemoveFromHook { m o }
  { \hook_gremove_code:nn {#1} {#2} }
```

(*End definition for* `\RemoveFromHook`. *This function is documented on page* 102.)

`\DeclareDefaultHookLabel`
`\__hook_curr_name_push:n`
`\__hook_curr_name_pop:`

The token list `\g__hook_hook_curr_name_tl` stores the name of the current package/file to be used as label for hooks. Providing a consistent interface is tricky, because packages can be loaded within packages, and some packages may not use `\DeclareDefaultHookLabel` to change the default label (in which case `\@currname` is used, if set).

To pull that off, we keep a stack that contains the default label for each level of input. The bottom of the stack contains the default label for the top-level. Since the string `top-level` is hardcoded, here this item of the stack is empty. This stack should never go empty.

But first, in case we're in latexrelease we push something on the stack to support roll forward. But in some rare cases, latexrelease may be loaded inside another package (notably platexrelease), so we'll dissect the `\@currnamestack`, adding empty items to the stack (empty so that defaults can kick in if needed):

```
⟨latexrelease⟩\cs_set_protected:Npn \__hook_tmp:w #1 #2 #3
⟨latexrelease⟩  {
⟨latexrelease⟩    \quark_if_recursion_tail_stop:n {#1}
⟨latexrelease⟩    \seq_gput_right:Nn \g__hook_name_stack_seq { }
⟨latexrelease⟩    \__hook_tmp:w
⟨latexrelease⟩  }
⟨latexrelease⟩\exp_after:wN \__hook_tmp:w \@currnamestack
⟨latexrelease⟩  \q_recursion_tail \q_recursion_tail
⟨latexrelease⟩  \q_recursion_tail \q_recursion_stop
```

Finally, an empty entry is added to the stack to account for the `top-level`. The item is empty so that the hard-coded default is used, but a call to `\DeclareDefaultHookLabel` will change it if needed.

```
\seq_gpush:Nn \g__hook_name_stack_seq { }
```

Two commands keep track of the stack: when a file is input, `\__hook_curr_name_push:n` pushes an (empty by default) label to the stack:

```
\cs_new_protected:Npn \__hook_curr_name_push:n #1
  {
    \seq_gpush:Nn \g__hook_name_stack_seq {#1}
    \tl_gset:Nn \g__hook_hook_curr_name_tl {#1}
  }
%
```

and when an input is over, the topmost item of the stack is popped, since the label will not be used again, and \g__hook_hook_curr_name_tl is updated to the now topmost item of the stack:

```
\cs_new_protected:Npn \__hook_curr_name_pop:
  {
    \seq_gpop:NN \g__hook_name_stack_seq \l__hook_return_tl
    \seq_get:NNTF \g__hook_name_stack_seq \l__hook_return_tl
      { \tl_gset_eq:NN \g__hook_hook_curr_name_tl \l__hook_return_tl }
      {
        \msg_error:nnn { hooks } { should-not-happen }
          { Tried~to~pop~from~an~empty~default~label~stack. }
      }
  }
```

The token list \g__hook_hook_curr_name_tl is but a mirror of the top of the stack.

Now define a wrapper that replaces the top of the stack with the argument, and updates \g__hook_hook_curr_name_tl accordingly.

```
\NewDocumentCommand \DeclareDefaultHookLabel { m }
  {
    \seq_gpop:NN \g__hook_name_stack_seq \l__hook_return_tl
    \__hook_curr_name_push:n {#1}
  }
```

(*End definition for* \DeclareDefaultHookLabel*,* __hook_curr_name_push:n*, and* __hook_curr_name_pop:*. This function is documented on page 104.*)

\UseHook
\UseOneTimeHook

Avoid the overhead of xparse and its protection that we don't want here (since the hook should vanish without trace if empty)!

```
\cs_new:Npn \UseHook        { \hook_use:n }
\cs_new:Npn \UseOneTimeHook { \hook_use_once:n }
```

(*End definition for* \UseHook *and* \UseOneTimeHook*. These functions are documented on page 101.*)

\ShowHook
\LogHook

```
\cs_new_protected:Npn \ShowHook { \hook_show:n }
\cs_new_protected:Npn \LogHook { \hook_log:n }
```

(*End definition for* \ShowHook *and* \LogHook*. These functions are documented on page 108.*)

\DebugHooksOn
\DebugHooksOff

```
\cs_new_protected:Npn \DebugHooksOn  { \hook_debug_on:  }
\cs_new_protected:Npn \DebugHooksOff { \hook_debug_off: }
```

(*End definition for* \DebugHooksOn *and* \DebugHooksOff*. These functions are documented on page 109.*)

\DeclareHookRule

```
\NewDocumentCommand \DeclareHookRule { m m m m }
                    { \hook_gset_rule:nnnn {#1}{#2}{#3}{#4} }
```

(*End definition for* \DeclareHookRule*. This function is documented on page 105.*)

\DeclareDefaultHookRule

This declaration is only supported before \begin{document}.

```
\NewDocumentCommand \DeclareDefaultHookRule { m m m }
                    { \hook_gset_rule:nnnn {??}{#1}{#2}{#3} }
\@onlypreamble\DeclareDefaultHookRule
```

(*End definition for* `\DeclareDefaultHookRule`. *This function is documented on page* 106.)

`\ClearHookRule` A special setup rule that removes an existing relation. Basically @@_rule_gclear:nnn plus fixing the property list for debugging.

FMi: Need an L3 interface, or maybe it should get dropped?

```
\NewDocumentCommand \ClearHookRule { m m m }
{ \hook_gset_rule:nnnn {#1}{#2}{unrelated}{#3} }
```

(*End definition for* `\ClearHookRule`. *This function is documented on page* 105.)

`\IfHookExistsTF` `\IfHookEmptyTF` Here we avoid the overhead of xparse, since `\IfHookEmptyTF` is used in `\end` (that is, every LaTeX environment). As a further optimisation, use `\let` rather than `\def` to avoid one expansion step.

```
\cs_new_eq:NN \IfHookExistsTF \hook_if_exist:nTF
\cs_new_eq:NN \IfHookEmptyTF \hook_if_empty:nTF
```

(*End definition for* `\IfHookExistsTF` *and* `\IfHookEmptyTF`. *These functions are documented on page* 107.)

5

3.13 Internal commands needed elsewhere

Here we set up a few horrible (but consistent) LaTeX 2_ε names to allow for internal commands to be used outside this module. We have to unset the `@@` since we want double "at" sign in place of double underscores.

```
⟨@@=⟩
```

`\@expl@@@initialize@all@@` `\@expl@@@hook@curr@name@push@@n` `\@expl@@@hook@curr@name@pop@@`

```
\cs_new_eq:NN \@expl@@@initialize@all@@
              \__hook_initialize_all:
\cs_new_eq:NN \@expl@@@hook@curr@name@push@@n
              \__hook_curr_name_push:n
\cs_new_eq:NN \@expl@@@hook@curr@name@pop@@
              \__hook_curr_name_pop:
```

(*End definition for* `\@expl@@@initialize@all@@`, `\@expl@@@hook@curr@name@push@@n`, *and* `\@expl@@@hook@curr@name@pop@@`. *These functions are documented on page* ??.)

```
\ExplSyntaxOff
```

Rolling back here doesn't undefine the interface commands as they may be used in packages without rollback functionality. So we just make them do nothing which may or may not work depending on the code usage.

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\NewHook}{The hook management}%
⟨latexrelease⟩
⟨latexrelease⟩\def\NewHook#1{}
⟨latexrelease⟩\def\NewReversedHook#1{}
⟨latexrelease⟩\def\NewMirroredHookPair#1#2{}
⟨latexrelease⟩
```

```
⟨latexrelease⟩\long\def\AddToHookNext#1#2{}
⟨latexrelease⟩
⟨latexrelease⟩\def\AddToHook#1{\@gobble@AddToHook@args}
⟨latexrelease⟩\providecommand\@gobble@AddToHook@args[2][]{}
⟨latexrelease⟩
⟨latexrelease⟩\def\RemoveFromHook#1{\@gobble@RemoveFromHook@arg}
⟨latexrelease⟩\providecommand\@gobble@RemoveFromHook@arg[1][]{}
⟨latexrelease⟩
⟨latexrelease⟩\def \UseHook        #1{}
⟨latexrelease⟩\def \UseOneTimeHook #1{}
⟨latexrelease⟩\def \ShowHook #1{}
⟨latexrelease⟩\let \DebugHooksOn \@empty
⟨latexrelease⟩\let \DebugHooksOff\@empty
⟨latexrelease⟩
⟨latexrelease⟩\def \DeclareHookRule #1#2#3#4{}
⟨latexrelease⟩\def \DeclareDefaultHookRule #1#2#3{}
⟨latexrelease⟩\def \ClearHookRule #1#2#3{}
```

If the hook management is not provided we make the test for existence false and the test for empty true in the hope that this is most of the time reasonable. If not a package would need to guard against running in an old kernel.

```
⟨latexrelease⟩\long\def \IfHookExistsTF #1#2#3{#3}
⟨latexrelease⟩\long\def \IfHookEmptyTF #1#2#3{#2}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

File h
ltalloc.dtx

1 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
```

(*End definition for* `\@xxxii`*. This function is documented on page* **??**.)

`\@Mi` `\@Mii` `\@Miii` `\@Miv` Constants 10001–10004.

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

(*End definition for* `\@Mi` *and others. These functions are documented on page* **??**.)

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

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

(*End definition for* `\@tempcnta` *and* `\@tempcntb`*. These functions are documented on page* **??**.)

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

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

(*End definition for* `\if@tempswa`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\@tempdima`, `\@tempdimb`, *and* `\@tempdimc`. *These functions are documented on page* **??**.)

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

```
\newbox\@tempboxa
```

(*End definition for* `\@tempboxa`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@tempskipa` *and* `\@tempskipb`. *These functions are documented on page* **??**.)

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

```
\newtoks\@temptokena
```

(*End definition for* `\@temptokena`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@flushglue`. *This function is documented on page* **??**.)

⟨/2ekernel⟩

File i
ltcntrl.dtx

1 Program control structure

This section defines a number of control structure macros, such as while-loops and for-loops.
Historical LaTeX 2.09 comments (not necessarily accurate any more):

```
⟨*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
```

End of historical LaTeX 2.09 comments.

\@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}}
```

(*End definition for* `\@whilenoop`, `\@whilenum`, *and* `\@iwhilenum`. *These functions are documented on page* ??.)

\@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}}
```

(*End definition for* `\@whiledim` *and* `\@iwhiledim`. *These functions are documented on page* ??.)

\@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}
```

(*End definition for* `\@whileswnoop`, `\@whilesw`, *and* `\@iwhilesw`. *These functions are documented on page* ??.)

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

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

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

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

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

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

End of historical LaTeX 2.09 comments.

`\@nnil`

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

(*End definition for* `\@nnil`. *This function is documented on page* ??.)

`\@empty`

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

(*End definition for* `\@empty`. *This function is documented on page* ??.)

\@fornoop

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

(*End definition for* `\@fornoop`. *This function is documented on page* ??.)

\@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}
```

(*End definition for* `\@for`. *This function is documented on page* ??.)

\@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}
```

(*End definition for* `\@forloop`. *This function is documented on page* ??.)

\@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}}
```

(*End definition for* `\@iforloop`. *This function is documented on page* ??.)

\@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}}
```

(*End definition for* `\@tfor`. *This function is documented on page* ??.)

\@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}
```

(*End definition for* `\@break@tfor`. *This function is documented on page* ??.)

\@removeelement Removes an element from a comma-separated list and puts it into a control sequence, called as `\@removeelement{`⟨*element*⟩`}{`⟨*list*⟩`}{`⟨*cs*⟩`}`. Due to the implementation method the ⟨*element*⟩ is not allowed to contain braces.

```
\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}}
```

(*End definition for* `\@removeelement`. *This function is documented on page* ??.)

⟨/2ekernel⟩

File j
lterror.dtx

1 Error handling and tracing

This section defines LaTeX's error commands.

```
⟨*2ekernel⟩
```

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

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

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

1.1 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
```

(*End definition for* `\MessageBreak`*. This function is documented on page* **??**.)

`\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
}
```

(*End definition for* `\GenericInfo`*. This function is documented on page* **??**.)

`\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
}
```

(*End definition for* `\GenericWarning`*. This function is documented on page* **??**.)

`\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.

```
\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}%
```

(*End definition for* `\GenericError`*. This function is documented on page* **??**.)

`\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.
```

```
\gdef\PackageError#1#2#3{%
   \GenericError{%
      (#1)\@spaces\@spaces\@spaces\@spaces
   }{%
      Package #1 Error: #2%
   }{%
      See the #1 package documentation for explanation.%
   }{#3}%
}
\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%
   }%
}
\gdef\ClassError#1#2#3{%
   \GenericError{%
      (#1) \space\@spaces\@spaces\@spaces
   }{%
      Class #1 Error: #2%
   }{%
      See the #1 class documentation for explanation.%
   }{#3}%
}
\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%
   }%
}
```

(*End definition for* `\PackageError` *and others. These functions are documented on page* **??**.)

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

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

```
\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}%
}
```

```
\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%
  }%
}
```

(*End definition for* `\@latex@error` *and others. These functions are documented on page* **??**.)

`\c@errorcontextlines` `\errorcontextlines` as a LaTeX counter, so that it may be manipulated with `\setcounter` (once it is defined :-)

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

(*End definition for* `\c@errorcontextlines`*. This function is documented on page* **??**.)

`\on@line` The message ' on input line n'.

```
\def\on@line{ on input line \the\inputlineno}
```

(*End definition for* `\on@line`*. This function is documented on page* **??**.)

`\@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
\global\let\@latexerr\@latex@error
```

(*End definition for* `\@warning`*,* `\@@warning`*, and* `\@latexerr`*. These functions are documented on page* **??**.)

`\@spaces` Four spaces.

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

(*End definition for* `\@spaces`*. This function is documented on page* **??**.)

1.2 Specific errors

`\@eha` `\@ehb` `\@ehc` `\@ehd` The more common error help messages.

```
\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}
```

(*End definition for* `\@eha` *and others. These functions are documented on page* **??**.)

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

(*End definition for* \@notdefinable*. This function is documented on page* ??.)

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

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

(*End definition for* \@nolnerr*. This function is documented on page* ??.)

\@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 2ε it MIGHT vanish! Use \@nocounterr{⟨*cnt*⟩} instead.

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

(*End definition for* \@nocounterr *and* \@nocnterr*. These functions are documented on page* ??.)

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

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

(*End definition for* \@ctrerr*. This function is documented on page* ??.)

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

```
\gdef\@nodocument{%
  \@latex@error{Missing \protect\begin{document}}\@ehd}
```

(*End definition for* \@nodocument*. This function is documented on page* ??.)

\@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{%
  \@latex@error{\protect\begin{\@currenvir}\@currenvline
                      \space ended by \protect\end{#1}}\@eha}
```

(*End definition for* \@badend*. This function is documented on page* ??.)

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

```
\gdef\@badmath{%
  \@latex@error{Bad math environment delimiter}\@eha}
```

(*End definition for* `\@badmath`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@toodeep`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@badpoptabs`. *This function is documented on page* **??**.)

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

```
\gdef\@badtab{%
 \@latex@error{Undefined tab position}\@ehd}
```

(*End definition for* `\@badtab`. *This function is documented on page* **??**.)

`\@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
    \@latex@error{\ifcase #1 Illegal character\or
     Missing @-exp\or Missing p-arg\fi\space
     in array arg}\@ehd
  \endgroup}
```

(*End definition for* `\@preamerr`. *This function is documented on page* **??**.)

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

```
\gdef\@badlinearg{%
  \@latex@error{%
       Bad \protect\line\space or \protect\vector
       \space argument}\@ehb}
```

(*End definition for* `\@badlinearg`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@parmoderr`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@fltovf`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@latexbug`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@badcrerr`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@noitemerr`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@notprerr`. *This function is documented on page* **??**.)

`\@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
    \@latex@error{Command \protect#1 invalid in math mode}\@ehc
   \fi}
```

(*End definition for* `\@inmatherr`. *This function is documented on page* **??**.)

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

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

(*End definition for* `\@invalidchar`. *This function is documented on page* **??**.)

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

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

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

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

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

1.3 Tracing

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

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

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

`\conditionally@traceon`
`\conditionally@traceoff`

These are only dummy definitions. For details see the `trace` package.

```
\let\conditionally@traceon\@empty
\let\conditionally@traceoff\@empty
```

(*End definition for* `\conditionally@traceon` *and* `\conditionally@traceoff`*. These functions are documented on page* ??.)

```
⟨/2ekernel⟩
```

File k
ltpar.dtx

1 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.

1.1 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
\@par

Initiate a long-term change to \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}
```

(*End definition for* \@setpar *and* \@par*. These functions are documented on page* ??.)

\@restorepar

Restore from a short-term change to \par.

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

(*End definition for* \@restorepar*. This function is documented on page* ??.)

File l
ltspace.dtx

1 Spacing

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

1.1 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.

1.2 Chris' comments

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

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

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

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

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

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

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

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

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

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

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

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

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

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

This is optimal coding since it is normal to index a word that gets split across a page-break on its starting page. This would, on the other hand, fix another common (and documented) failure of the current system: when the invisible is the last thing in a paragraph the space before it is not removed and, worse, it is also hidden from the paragraph-ending mechanism so that an 'empty' line can be created at the end of the paragraph.

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

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

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

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

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

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

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

```
... end of text.

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

1.3 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(?) `\unskips` include test for zero skip (rather than other tests or no test).
- Consider replacing `\hskip 1sp` by something better (here called an 'infinitesimal' skip).
- Look at all `\hskip\z@` (or similar) to see if they should be changed to an 'infinitesimal' skip.

- Resolve the inconsistency between `\hspace` and `\hspace*`.
- Remove unnecessary `\unskips`.
- Investigate and rationalise the 'newline' code.
- Find better algorithms for all sorts of things or, easier(?), fix TEX itself.

1.4 The code

```
⟨*2ekernel⟩
\message{spacing,}
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\pagebreak}{Make commands robust}%
```

`\pagebreak`
`\nopagebreak`

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

(*End definition for* `\pagebreak` *and* `\nopagebreak`*. These functions are documented on page* **??**.)

`\linebreak`
`\nolinebreak`

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

(*End definition for* `\linebreak` *and* `\nolinebreak`*. These functions are documented on page* **??**.)

`\samepage`

```
\DeclareRobustCommand\samepage{\interlinepenalty\@M
   \postdisplaypenalty\@M
   \interdisplaylinepenalty\@M
   \@beginparpenalty\@M
   \@endparpenalty\@M
   \@itempenalty\@M
   \@secpenalty\@M
   \interfootnotelinepenalty\@M}
```

(*End definition for* `\samepage`*. This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\pagebreak}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\pagebreak
⟨latexrelease⟩\kernel@make@fragile\nopagebreak
⟨latexrelease⟩\kernel@make@fragile\linebreak
⟨latexrelease⟩\kernel@make@fragile\nolinebreak
⟨latexrelease⟩\kernel@make@fragile\samepage
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

\@no@pgbk

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

(*End definition for* `\@no@pgbk`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* `\@no@lnbk`*. This function is documented on page* **??**.)

\\ 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 `\\`.

The internal definition of the 'normal' definition of `\\`.

\@normalcr

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\@normalcr}{Make robust}%
\protected\def\@normalcr{%
  \let \reserved@e \relax
  \let \reserved@f \relax
  \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
             \@xnewline}%
  \@xnewline}
```

```
\let\\\@normalcr
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@normalcr}{Make robust}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\\{%
⟨latexrelease⟩   \let \reserved@e \relax
⟨latexrelease⟩   \let \reserved@f \relax
⟨latexrelease⟩   \@ifstar{\let \reserved@e \vadjust \let \reserved@f \nobreak
⟨latexrelease⟩             \@xnewline}%
⟨latexrelease⟩  \@xnewline}
⟨latexrelease⟩\expandafter\let\expandafter\@normalcr
⟨latexrelease⟩     \csname\expandafter\@gobble\string\\ \endcsname
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\\` *and* `\@normalcr`*. These functions are documented on page* **??**.)

`\@vspace@calcify` Helper command to produce a `\vskip` that is first run through `\setlength`. This way the `calc` package can operate on the argument value.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@vspace@calcify}{Add calc support}%
\def\@vspace@calcify#1{\begingroup\setlength\skip@{#1}\vskip\skip@\endgroup}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@vspace@calcify}{Add calc support}%
⟨latexrelease⟩
⟨latexrelease⟩\let\@vspace@calcify\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@vspace@calcify`*. This function is documented on page* **??**.)

`\newline` A simple form of the 'normal' definition of `\\`.

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

(*End definition for* `\newline`*. This function is documented on page* **??**.)

`\@xnewline`

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

(*End definition for* `\@xnewline`*. This function is documented on page* **??**.)

`\@newline`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
```

```
⟨latexrelease⟩                  {\@newline}{\newline calc support}%
\def\@newline[#1]{\let \reserved@e \vadjust
                    \@gnewline {\@vspace@calcify{#1}}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@newline}{\newline calc support}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@newline[#1]{\let \reserved@e \vadjust
⟨latexrelease⟩                    \@gnewline {\vskip #1}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@newline`*. This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@gnewline`*. This function is documented on page* **??**.)

`\@getpen`

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

(*End definition for* `\@getpen`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\if@nobreak`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\@savsk` *and* `\@savsf`*. These functions are documented on page* **??**.)

\@bsphack

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

Before giving the current definition we give an extended definition that is currently not used (because it doesn't work as advertised:-)

These are generalised hacks which attempt to do sensible things when 'invisible commands' appear in vmode too.

They need to cope with space in both hmode (plus spacefactor) and vmode, and also cope with breaks etc. In vmode this means ensuring that any following \addvspace, etc sees the correct glue in \lastskip.

In fact, these improved versions should be used for other cases of 'whatsits, thingies etc' which should be invisible. They are only for commands, not environments (see notes on \@Esphack).

BTW, anyone know why the standard hacks are surrounded by \ifmmode\else rather than simply \ifhmode?

And are there any cases where saving the spacefactor is essential? I have some extensions where it is, but it does not appear to be so in the standard uses.

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

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

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

(*End definition for* `\@bsphack`*. This function is documented on page* ??.)

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

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2018/10/10}%
⟨latexrelease⟩                  {\@esphack}{hyphenation and nobreak after space hack}%
⟨*2ekernel | latexrelease⟩
\def\@esphack{%
  \relax
  \ifhmode
    \spacefactor\@savsf
    \ifdim\@savsk>\z@

      \ifdim\lastskip=\z@
        \nobreak \hskip\z@skip
      \fi
      \ignorespaces
    \fi

  \else
    \ifvmode
      \if@nobreak\nobreak\else\if@noskipsec\nobreak\fi\fi
    \fi

  \fi}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/10/01}%
⟨latexrelease⟩                  {\@esphack}{hyphenation and nobreak after space hack}%
⟨latexrelease⟩\def\@esphack{%
⟨latexrelease⟩  \relax
⟨latexrelease⟩  \ifhmode
⟨latexrelease⟩    \spacefactor\@savsf
⟨latexrelease⟩    \ifdim\@savsk>\z@
⟨latexrelease⟩      \ifdim\lastskip=\z@
⟨latexrelease⟩        \nobreak \hskip\z@skip
⟨latexrelease⟩      \fi
```

```
⟨latexrelease⟩      \ignorespaces
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                 {\@esphack}{hyphenation and nobreak after space hack}%
⟨latexrelease⟩\def\@esphack{%
⟨latexrelease⟩  \relax
⟨latexrelease⟩  \ifhmode
⟨latexrelease⟩    \spacefactor\@savsf
⟨latexrelease⟩    \ifdim\@savsk>\z@
⟨latexrelease⟩      \nobreak \hskip\z@skip
⟨latexrelease⟩      \ignorespaces
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@esphack}{hyphenation and nobreak after space hack}%
⟨latexrelease⟩\def\@esphack{%
⟨latexrelease⟩  \relax
⟨latexrelease⟩  \ifhmode
⟨latexrelease⟩    \spacefactor\@savsf
⟨latexrelease⟩    \ifdim\@savsk>\z@
⟨latexrelease⟩      \ignorespaces
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@esphack`*. This function is documented on page* **??**.)

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

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                 {\@Esphack}{hyphenation after space hack}%
⟨*2ekernel | latexrelease⟩
\def\@Esphack{%
  \relax
  \ifhmode
    \spacefactor\@savsf
    \ifdim\@savsk>\z@
      \nobreak \hskip\z@skip
      \@ignoretrue
      \ignorespaces
    \fi
   \fi}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@Esphack}{hyphenation after space hack}%
⟨latexrelease⟩\def\@Esphack{%
⟨latexrelease⟩  \relax
⟨latexrelease⟩  \ifhmode
```

```
⟨latexrelease⟩     \spacefactor\@savsf
⟨latexrelease⟩     \ifdim\@savsk>\z@
⟨latexrelease⟩       \@ignoretrue
⟨latexrelease⟩       \ignorespaces
⟨latexrelease⟩     \fi
⟨latexrelease⟩    \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@Esphack`*. This function is documented on page* **??**.)

`\@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
}
```

(*End definition for* `\@vbsphack`*. This function is documented on page* **??**.)

1.5 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}
```

(*End definition for* `\@xaddvskip`*. This function is documented on page* ??.)

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

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\addvspace}{\addvspace calc support}%
\def\addvspace#1{%
  \ifvmode
     \if@minipage\else
       \ifdim \lastskip =\z@
         \@vspace@calcify{#1}%
       \else
       \setlength\@tempskipb{#1}%
         \@xaddvskip
       \fi
     \fi
  \else
    \@noitemerr
  \fi}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\addvspace}{\addvspace calc support}%
```

```
⟨latexrelease⟩
⟨latexrelease⟩\def\addvspace#1{%
⟨latexrelease⟩  \ifvmode
⟨latexrelease⟩     \if@minipage\else
⟨latexrelease⟩       \ifdim \lastskip =\z@
⟨latexrelease⟩         \vskip #1\relax
⟨latexrelease⟩       \else
⟨latexrelease⟩       \@tempskipb#1\relax
⟨latexrelease⟩         \@xaddvskip
⟨latexrelease⟩       \fi
⟨latexrelease⟩     \fi
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@noitemerr
⟨latexrelease⟩  \fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\addvspace`*. This function is documented on page* **??**.)

`\addpenalty`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\addpenalty}{\addpenalty}%
⟨*2ekernel | latexrelease⟩
```

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

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

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

```
          \begingroup
            \@tempskipa\@tempskipb
            \advance \@tempskipb
              \ifdim\prevdepth>\maxdepth\maxdepth\else
```

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

```
                \ifdim \prevdepth = -\@m\p@ \z@ \else \prevdepth \fi
              \fi
            \vskip -\@tempskipb
            \penalty#1%
            \ifdim\@tempskipa=\@tempskipb
```

Do nothing if the \prevdepth check made no adjustment.

```
            \else
```

Combine the prevdepth adjustment into a single skip.

```
              \advance\@tempskipb -\@tempskipa
              \vskip \@tempskipb
            \fi
```

The final skip is always the specified length.

```
            \vskip \@tempskipa
          \endgroup
        \fi
      \fi
    \fi
  \else
    \@noitemerr
  \fi}%
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\addpenalty}{\addpenalty}%
⟨latexrelease⟩\def\addpenalty#1{%
⟨latexrelease⟩  \ifvmode
⟨latexrelease⟩    \if@minipage
⟨latexrelease⟩    \else
⟨latexrelease⟩      \if@nobreak
⟨latexrelease⟩      \else
⟨latexrelease⟩        \ifdim\lastskip=\z@
⟨latexrelease⟩          \penalty#1\relax
⟨latexrelease⟩        \else
⟨latexrelease⟩          \@tempskipb\lastskip
⟨latexrelease⟩          \vskip -\lastskip
⟨latexrelease⟩          \penalty#1%
⟨latexrelease⟩          \vskip\@tempskipb
⟨latexrelease⟩        \fi
⟨latexrelease⟩      \fi
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@noitemerr
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \addpenalty*. This function is documented on page* ??.)

\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}
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@vspace}{Support calc in \vspace}%
```

We support calc syntax in the argument and therefore use `\setlength`.

```
\def\@vspace #1{%
  \ifvmode
    \@vspace@calcify{#1}%
    \vskip\z@skip
   \else
     \@bsphack
     \vadjust{\@restorepar
              \@vspace@calcify{#1}%
              \vskip\z@skip
              }%
     \@esphack
   \fi}
```

```
\def\@vspacer#1{%
  \ifvmode
    \dimen@\prevdepth
    \hrule \@height\z@
    \nobreak
    \@vspace@calcify{#1}%
    \vskip\z@skip
    \prevdepth\dimen@
  \else
    \@bsphack
    \vadjust{\@restorepar
             \hrule \@height\z@
             \nobreak
             \@vspace@calcify{#1}%
             \vskip\z@skip}%
    \@esphack
  \fi}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@vspace}{Support calc in \vspace}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@vspace #1{%
⟨latexrelease⟩  \ifvmode
⟨latexrelease⟩    \vskip #1
⟨latexrelease⟩    \vskip\z@skip
⟨latexrelease⟩   \else
⟨latexrelease⟩     \@bsphack
⟨latexrelease⟩     \vadjust{\@restorepar
⟨latexrelease⟩              \vskip #1
⟨latexrelease⟩              \vskip\z@skip
⟨latexrelease⟩              }%
⟨latexrelease⟩     \@esphack
⟨latexrelease⟩   \fi}
⟨latexrelease⟩\def\@vspacer#1{%
⟨latexrelease⟩  \ifvmode
```

```
⟨latexrelease⟩    \dimen@\prevdepth
⟨latexrelease⟩    \hrule \@height\z@
⟨latexrelease⟩    \nobreak
⟨latexrelease⟩    \vskip #1
⟨latexrelease⟩    \vskip\z@skip
⟨latexrelease⟩    \prevdepth\dimen@
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@bsphack
⟨latexrelease⟩    \vadjust{\@restorepar
⟨latexrelease⟩             \hrule \@height\z@
⟨latexrelease⟩             \nobreak
⟨latexrelease⟩             \vskip #1
⟨latexrelease⟩             \vskip\z@skip}%
⟨latexrelease⟩    \@esphack
⟨latexrelease⟩  \fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\vspace`, `\@vspace`, *and* `\@vspacer`. *These functions are documented on page* **??**.)

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

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

(*End definition for* `\smallskip`, `\medskip`, *and* `\bigskip`. *These functions are documented on page* **??**.)

`\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
```

(*End definition for* `\smallskipamount`, `\medskipamount`, *and* `\bigskipamount`. *These functions are documented on page* **??**.)

1.6 Horizontal space (and breaks)

`\nobreakdashes`

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

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

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

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

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

```
\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
}
```

(*End definition for* `\nobreakdashes`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\nobreakspace` *and* `\@xobeysp`*. These functions are documented on page* **??**.)

`\@`

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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\@}{Space after \@}%
⟨*2ekernel | latexrelease⟩
\def\@{\spacefactor\@m{}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@}{Space after \@}%
⟨latexrelease⟩\def\@{\spacefactor\@m}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@`*. This function is documented on page* **??**.)

`\hspace`

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

(*End definition for* `\hspace`*. This function is documented on page* **??**.)

\@hspace

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@hspace}{Support calc with \hspace}%
\def\@hspace#1{\begingroup\setlength\skip@{#1}\hskip\skip@\endgroup}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@hspace}{Support calc with \hspace}%

⟨latexrelease⟩
⟨latexrelease⟩\def\@hspace#1{\hskip #1\relax}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@hspace`*. This function is documented on page* **??**.)

\@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
                \@hspace{#1}\hskip \z@skip}
```

(*End definition for* `\@hspacer`*. This function is documented on page* **??**.)

\fill

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

(*End definition for* `\fill`*. This function is documented on page* **??**.)

\stretch

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

(*End definition for* `\stretch`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\thinspace}{Start LR-mode}%
```

\enspace

```
\DeclareRobustCommand\enspace{\leavevmode@ifvmode\kern.5em }
```

(*End definition for* `\enspace`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\leavevmode@ifvmode`. *This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\thinspace}{Start LR-mode}%
⟨latexrelease⟩\def\thinspace{\kern .16667em }
⟨latexrelease⟩\def\negthinspace{\kern-.16667em }
⟨latexrelease⟩\def\enspace{\kern.5em }
⟨latexrelease⟩\let\leavevmode@ifvmode\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\enskip`
`\quad`
`\qquad`

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

(*End definition for* `\enskip`, `\quad`, *and* `\qquad`. *These functions are documented on page* **??**.)

For Unicode engines, make the Unicode soft hyphen an active character defined as `\-`.

```
\ifx\Umathcode\@undefined\else
  \catcode "AD=13
  \def^^ad{\-}
\fi
```

`\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 }}
```

(*End definition for* `\obeycr` *and* `\restorecr`. *These functions are documented on page* **??**.)

```
⟨/2ekernel⟩
```

File m
ltlogos.dtx

1 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⟩
\DeclareRobustCommand\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
```

(*End definition for* \TeX*. This function is documented on page* ??.)

\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}
```

(*End definition for* \LaTeX*. This function is documented on page* ??.)

\LaTeXe The LATEX 2_ε 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⟩
```

(*End definition for* \LaTeXe*. This function is documented on page* ??.)

File n
ltfiles.dtx

1 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*.
Historical LaTeX 2.09 comments (not necessarily accurate any more):

```
⟨*2ekernel⟩
\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

End of historical LaTeX 2.09 comments.

\@inputcheck
\@unused

Allocate read stream for testing and output stream.

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

(*End definition for* `\@inputcheck` *and* `\@unused`*. These functions are documented on page* **??**.)

\@mainaux
\@partaux

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

(*End definition for* `\@mainaux` *and* `\@partaux`*. These functions are documented on page* **??**.)

\if@filesw
\if@partsw

```
\newif\if@filesw \@fileswtrue
\newif\if@partsw \@partswfalse
```

(*End definition for* `\if@filesw` *and* `\if@partsw`*. These functions are documented on page* **??**.)

\@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
```

(*End definition for* `\@clubpenalty`*. This function is documented on page* **??**.)

\document

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩  {\document}{Added hook to load l3backend code}%
⟨*2ekernel | latexrelease⟩
\def\document{%
```

We do cancel the grouping as part of the `\begin` handling (this is now done inside `\begin` instead) so that the `env/⟨env⟩/begin` hook is not hidden inside `\begingroup ... \endgroup`.

```
% \endgroup
  \UseOneTimeHook{begindocument/before}%
  \@kernel@after@begindocument@before
```

Added hook to load l3backend code:

```
\@expl@sys@load@backend@@
\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
```

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

```
\ifx\document@default@language\m@ne
  \chardef\document@default@language\language
\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...

```
\@kernel@before@begindocument
\UseOneTimeHook{begindocument}%
\@kernel@after@begindocument
```

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
```

```
\UseOneTimeHook{begindocument/end}%
```

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

```
\ignorespaces}
```

The begindocument hook already existed in the kernel since 1994 under the name \atbegindocumenthook the additional ones are originally from the etoolbox package under the names \@endpreamblehook \afterpreamble.

```
\NewHook{begindocument}
\NewHook{begindocument/before}
\NewHook{begindocument/end}
```

\@kernel@after@begindocument@before
\@kernel@before@begindocument
\@kernel@after@begindocument

Above we used two kernel only hooks to be run after the public `begindocument/before` and after `begindocument` hooks.

In `\@kernel@after@begindocument@before` we already place one action: drop the fast execution code for the `env/document/begin` hook. That hook marks the end of the preamble and should therefore only be run once. In a normal document that is anyway the case (so the code would just sit there taking up space afterwards, which these days is rather harmless), however, in more complicated scenarios where several full documents are combined to a single document it might get applied several times with harmful effects. We therefore explicitly drop it at this point. the coing is somewhat obscure due to the name of the macro which requires constructing.

```
\edef \@kernel@after@begindocument@before {%
  \let\expandafter\noexpand\csname
       __hook env/document/begin\endcsname
  \noexpand\@empty}
```

```
\let \@kernel@before@begindocument \@empty
\let \@kernel@after@begindocument \@empty
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{2017/04/15}%
⟨latexrelease⟩  {\document}{Save language for hyphenation}%
⟨latexrelease⟩
⟨latexrelease⟩\def\document{\endgroup
⟨latexrelease⟩  \ifx\@unusedoptionlist\@empty\else
⟨latexrelease⟩    \@latex@warning@no@line{Unused global option(s):^^J%
⟨latexrelease⟩            \@spaces[\@unusedoptionlist]}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@colht\textheight
⟨latexrelease⟩  \@colroom\textheight \vsize\textheight
⟨latexrelease⟩  \columnwidth\textwidth
⟨latexrelease⟩  \@clubpenalty\clubpenalty
⟨latexrelease⟩  \if@twocolumn
⟨latexrelease⟩    \advance\columnwidth -\columnsep
⟨latexrelease⟩    \divide\columnwidth\tw@ \hsize\columnwidth \@firstcolumntrue
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \hsize\columnwidth \linewidth\hsize
⟨latexrelease⟩  \begingroup\@floatplacement\@dblfloatplacement
⟨latexrelease⟩    \makeatletter\let\@writefile\@gobbletwo
⟨latexrelease⟩    \global \let \@multiplelabels \relax
⟨latexrelease⟩    \@input{\jobname.aux}%
⟨latexrelease⟩  \endgroup
⟨latexrelease⟩  \if@filesw
⟨latexrelease⟩    \immediate\openout\@mainaux\jobname.aux
⟨latexrelease⟩    \immediate\write\@mainaux{\relax}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \process@table
⟨latexrelease⟩  \let\glb@currsize\@empty  % Force math initialization.
⟨latexrelease⟩  \normalsize
⟨latexrelease⟩  \everypar{}%
⟨latexrelease⟩  \ifx\normalsfcodes\@empty
⟨latexrelease⟩    \ifnum\sfcode`\.=\@m
⟨latexrelease⟩      \let\normalsfcodes\frenchspacing
```

```
⟨latexrelease⟩    \else
⟨latexrelease⟩      \let\normalsfcodes\nonfrenchspacing
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \ifx\document@default@language\m@ne
⟨latexrelease⟩    \chardef\document@default@language\language
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@noskipsecfalse
⟨latexrelease⟩  \let \@refundefined \relax
⟨latexrelease⟩  \let\AtBeginDocument\@firstofone
⟨latexrelease⟩  \@begindocumenthook
⟨latexrelease⟩  \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
⟨latexrelease⟩  \global\@maxdepth\maxdepth
⟨latexrelease⟩  \global\let\@begindocumenthook\@undefined
⟨latexrelease⟩  \ifx\@listfiles\@undefined
⟨latexrelease⟩    \global\let\@filelist\relax
⟨latexrelease⟩    \global\let\@addtofilelist\@gobble
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \gdef\do##1{\global\let ##1\@notprerr}%
⟨latexrelease⟩  \@preamblecmds
⟨latexrelease⟩  \global\let \@nodocument \relax
⟨latexrelease⟩  \global\let\do\noexpand
⟨latexrelease⟩  \ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\document}{Save language for hyphenation}
⟨latexrelease⟩\def\document{\endgroup
⟨latexrelease⟩  \ifx\@unusedoptionlist\@empty\else
⟨latexrelease⟩    \@latex@warning@no@line{Unused global option(s):^^J%
⟨latexrelease⟩            \@spaces[\@unusedoptionlist]}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@colht\textheight
⟨latexrelease⟩  \@colroom\textheight \vsize\textheight
⟨latexrelease⟩  \columnwidth\textwidth
⟨latexrelease⟩  \@clubpenalty\clubpenalty
⟨latexrelease⟩  \if@twocolumn
⟨latexrelease⟩    \advance\columnwidth -\columnsep
⟨latexrelease⟩    \divide\columnwidth\tw@ \hsize\columnwidth
⟨latexrelease⟩    \@firstcolumntrue
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \hsize\columnwidth \linewidth\hsize
⟨latexrelease⟩  \begingroup\@floatplacement\@dblfloatplacement
⟨latexrelease⟩    \makeatletter\let\@writefile\@gobbletwo
⟨latexrelease⟩    \global \let \@multiplelabels \relax
⟨latexrelease⟩    \@input{\jobname.aux}%
⟨latexrelease⟩  \endgroup
⟨latexrelease⟩  \if@filesw
⟨latexrelease⟩    \immediate\openout\@mainaux\jobname.aux
⟨latexrelease⟩    \immediate\write\@mainaux{\relax}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \process@table
⟨latexrelease⟩  \let\glb@currsize\@empty
⟨latexrelease⟩  \normalsize
```

```
⟨latexrelease⟩  \everypar{}%
⟨latexrelease⟩  \ifx\normalsfcodes\@empty
⟨latexrelease⟩    \ifnum\sfcode`\.=\@m
⟨latexrelease⟩      \let\normalsfcodes\frenchspacing
⟨latexrelease⟩    \else
⟨latexrelease⟩      \let\normalsfcodes\nonfrenchspacing
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@noskipsecfalse
⟨latexrelease⟩  \let \@refundefined \relax
⟨latexrelease⟩  \let\AtBeginDocument\@firstofone
⟨latexrelease⟩  \@begindocumenthook
⟨latexrelease⟩  \ifdim\topskip<1sp\global\topskip 1sp\relax\fi
⟨latexrelease⟩  \global\@maxdepth\maxdepth
⟨latexrelease⟩  \global\let\@begindocumenthook\@undefined
⟨latexrelease⟩  \ifx\@listfiles\@undefined
⟨latexrelease⟩    \global\let\@filelist\relax
⟨latexrelease⟩    \global\let\@addtofilelist\@gobble
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \gdef\do##1{\global\let ##1\@notprerr}%
⟨latexrelease⟩  \@preamblecmds
⟨latexrelease⟩  \global\let \@nodocument \relax
⟨latexrelease⟩  \global\let\do\noexpand
⟨latexrelease⟩  \ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\document
```

(*End definition for* `\document` *and others. These functions are documented on page* ??.)

`\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
```

(*End definition for* `\normalsfcodes`*. This function is documented on page* ??.)

`\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
```

(*End definition for* `\nofiles`*. This function is documented on page* ??.)

\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
}
```

(*End definition for* \protected@write*. This function is documented on page* ??.)

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

\include \includeonly 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.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\includeonly}{Spaces in file names}%
```

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

Here the normalisation will add .tex for all files, (it uses the the same normalisation as the hooks), so we need to remove that manually. \@strip@tex@ext does that.

```
    \set@curr@file{#1}%
    \edef\@curr@file{\@strip@tex@ext\@curr@file}%
```

For historical reasons \@include expects an argument delimited by a space. This is kept (though uncessary now) to avoid errors in other packages that use \@include directly.

```
    \expandafter\@include\expandafter{\@curr@file} % deliberate space
  \fi}
```

Here in \includeonly we also need to strip .tex after normalisation:

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

Because the argument to \includeonly is a comma-separated list of filenames where there may be comma's precedeing some of the filenames or trailing them. Therefore we need to take the list apart, remove the unwanted spaces while leaving the spaces *in* the filenames intact.

```
  \let\@partlist\@empty
  \@for\reserved@a:=#1 \do
```

```
    {%
      \expandafter\set@curr@file\expandafter{\reserved@a}%
      \ifx\@partlist\@empty
        \edef\@partlist{\@strip@tex@ext\@curr@file}%
      \else
        \edef\@partlist{\@partlist,\@strip@tex@ext\@curr@file}%
      \fi
    }%
  }
\@onlypreamble\includeonly
```

(*End definition for* `\include` *and* `\includeonly`*. These functions are documented on page* ??.)

`\@strip@tex@ext`
`\@strip@tex@ext@aux`

These macros take a (`\detokenize`d file name and remove any `.tex` extension). Extra care is taken to not remove the string `.tex` from the middle of a file name: it is only removed if it's the very last thing in the file name.

```
\def\reserved@a#1{%
\def\@strip@tex@ext##1{%
  \expandafter\@strip@tex@ext@aux
    ##1\@nil\@nil
     #1\@nil\relax\@nnil}
\def\@strip@tex@ext@aux##1#1\@nil##2\@nnil{%
  \ifx\relax##2\@empty
    \expandafter\@cdr\expandafter\@empty\@cdr{}##1%
  \else##1\fi}}%
\expandafter\reserved@a
\expandafter{\detokenize{.tex}}
⟨/2ekernel | latexrelease⟩
```

(*End definition for* `\@strip@tex@ext` *and* `\@strip@tex@ext@aux`*. These functions are documented on page* ??.)

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\includeonly}{Spaces in file names}%
⟨latexrelease⟩
⟨latexrelease⟩\def\includeonly#1{%
⟨latexrelease⟩  \@partswtrue
⟨latexrelease⟩  \set@curr@file{\zap@space#1 \@empty}%
⟨latexrelease⟩  \let\@partlist\@curr@file
⟨latexrelease⟩  }
⟨latexrelease⟩
⟨latexrelease⟩\def\include#1{\relax
⟨latexrelease⟩  \ifnum\@auxout=\@partaux
⟨latexrelease⟩    \@latex@error{\string\include\space cannot be nested}\@eha
⟨latexrelease⟩  \else
⟨latexrelease⟩    \set@curr@file{#1 }%
⟨latexrelease⟩    \expandafter\@include\@curr@file
⟨latexrelease⟩  \fi}
⟨latexrelease⟩
⟨latexrelease⟩\let\@strip@tex@ext\@undefined
⟨latexrelease⟩\let\@strip@tex@ext@aux\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\includeonly}{Spaces in file names}%
⟨latexrelease⟩\def\includeonly#1{%
⟨latexrelease⟩   \@partswtrue
⟨latexrelease⟩   \edef\@partlist{\zap@space#1 \@empty}}
⟨latexrelease⟩
⟨latexrelease⟩\def\include#1{\relax
⟨latexrelease⟩   \ifnum\@auxout=\@partaux
⟨latexrelease⟩     \@latex@error{\string\include\space cannot be nested}\@eha
⟨latexrelease⟩   \else \@include#1 \fi}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\@include`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@include}{Spaces in file names and hooks}%
```

```
\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
```

Now before going to the hooks we need to set `\CurrentFile`:

```
%----------------------------------------
    \@filehook@set@CurrentFile
```

Execute the `before` hooks just after we switched the `.aux` file ...

```
    \UseHook{include/before}%
    \UseHook{include/before/#1}%
%----------------------------------------
    \@input@{#1.tex}%
%----------------------------------------
```

... then `end` hooks ...

```
    \UseHook{include/end/#1}%
    \UseHook{include/end}%
%----------------------------------------
    \clearpage
%----------------------------------------
```

... and after the `\clearpage` the `after` hooks followed by another `\clearpage` just in case new material got added (after all we need to be in well defined state after the `\include`).

```
    \UseHook{include/after/#1}%
    \UseHook{include/after}%
    \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}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@include}{Spaces in file names}%
⟨latexrelease⟩\def\@include#1 {%
⟨latexrelease⟩  \clearpage
⟨latexrelease⟩  \if@filesw
⟨latexrelease⟩    \immediate\write\@mainaux{\string\@input{#1.aux}}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@tempswatrue
⟨latexrelease⟩  \if@partsw
⟨latexrelease⟩    \@tempswafalse
⟨latexrelease⟩    \edef\reserved@b{#1}%
⟨latexrelease⟩    \@for\reserved@a:=\@partlist\do
⟨latexrelease⟩      {\ifx\reserved@a\reserved@b\@tempswatrue\fi}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@tempswa
⟨latexrelease⟩    \let\@auxout\@partaux
⟨latexrelease⟩    \if@filesw
⟨latexrelease⟩      \immediate\openout\@partaux #1.aux
⟨latexrelease⟩      \immediate\write\@partaux{\relax}%
⟨latexrelease⟩    \fi
⟨latexrelease⟩    \@input@{#1.tex}%
⟨latexrelease⟩    \clearpage
⟨latexrelease⟩    \@writeckpt{#1}%
⟨latexrelease⟩    \if@filesw
⟨latexrelease⟩      \immediate\closeout\@partaux
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \else
⟨latexrelease⟩    \deadcycles\z@
⟨latexrelease⟩    \@nameuse{cp@#1}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \let\@auxout\@mainaux}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨*2ekernel⟩
```

(*End definition for* `\@include`*. This function is documented on page* **??**.)

`\@writeckpt`

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

(*End definition for* `\@writeckpt`*. This function is documented on page* **??**.)

`\@wckptelt`

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

(*End definition for* `\@wckptelt`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\@setckpt`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@charlb` *and* `\@charrb`*. These functions are documented on page* **??**.)

1.1 Safe Input Macros

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

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

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

The definition below is from 2019 and only used during kernel bootstrapping, later on in `ltfilehook.dtx` it will get overwritten.

```
\def\set@curr@file#1{%
  \begingroup
    \escapechar\m@ne
    \xdef\@curr@file{%
      \expandafter\expandafter\expandafter\unquote@name
      \expandafter\expandafter\expandafter{%
      \expandafter\string
        \csname\@firstofone#1\@empty\endcsname}}%
  \endgroup
}
```

(*End definition for* `\@curr@file` *and* `\set@curr@file`*. These functions are documented on page* **??**.)

`\quote@name`
`\quote@@name`
`\unquote@name`

Quoting spaces

```
a b c     -> "a b c"
"a b c"   -> "a b c"
a" "b" "c -> "a b c"
          -> ""
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\quote@name}{Quote file names}%
\def\quote@name#1{"\quote@@name#1\@gobble""}
\def\quote@@name#1"{#1\quote@@name}
```

and removing quotes ...

```
\def\unquote@name#1{\quote@@name#1\@gobble"}
```

(*End definition for* `\quote@name`*,* `\quote@@name`*, and* `\unquote@name`*. These functions are documented on page* **??**.)

`\IfFileExists`

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

(*End definition for* `\IfFileExists`*. This function is documented on page* **??**.)

`\IfFileExists@`

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

```
\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}
```

(*End definition for* `\IfFileExists@`. *This function is documented on page* **??**.)

`\@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\expandafter\quote@name\expandafter{\reserved@b#1} %
    \ifeof\@inputcheck\else
      \edef\@filef@und{\expandafter\quote@name\expandafter{\reserved@b#1} }%
      \let\reserved@a\@firstoftwo%
      \closein\@inputcheck
      \@break@tfor
    \fi}%
  \reserved@a}
```

(*End definition for* `\@iffileonpath`. *This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\quote@name}{Quote file names}%
⟨latexrelease⟩
⟨latexrelease⟩\let\quote@name\@undefined
⟨latexrelease⟩\let\quote@@name\@undefined
⟨latexrelease⟩\let\unquote@name\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\IfFileExists@\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\long\def \IfFileExists#1#2#3{%
⟨latexrelease⟩  \openin\@inputcheck#1 %
⟨latexrelease⟩  \ifeof\@inputcheck
⟨latexrelease⟩    \ifx\input@path\@undefined
⟨latexrelease⟩      \def\reserved@a{#3}%
⟨latexrelease⟩    \else
⟨latexrelease⟩      \def\reserved@a{\@iffileonpath{#1}{#2}{#3}}%
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \else
⟨latexrelease⟩    \closein\@inputcheck
⟨latexrelease⟩    \edef\@filef@und{#1 }%
⟨latexrelease⟩    \def\reserved@a{#2}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \reserved@a}
⟨latexrelease⟩
⟨latexrelease⟩\long\def\@iffileonpath#1{%
⟨latexrelease⟩  \let\reserved@a\@secondoftwo
⟨latexrelease⟩  \expandafter\@tfor\expandafter\reserved@b\expandafter
⟨latexrelease⟩             :\expandafter=\input@path\do{%
⟨latexrelease⟩    \openin\@inputcheck\reserved@b#1 %
⟨latexrelease⟩    \ifeof\@inputcheck\else
⟨latexrelease⟩      \edef\@filef@und{\reserved@b#1 }%
⟨latexrelease⟩      \let\reserved@a\@firstoftwo%
⟨latexrelease⟩      \closein\@inputcheck
⟨latexrelease⟩      \@break@tfor
```

```
⟨latexrelease⟩     \fi}%
⟨latexrelease⟩   \reserved@a}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

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

This here is a temporary definition for the kernel. The real one comes somewhat later in the file `ltfilehook.dtx`.

```
\DeclareRobustCommand \InputIfFileExists[2]{%
  \IfFileExists{#1}%
    {%
  \expandafter\@swaptwoargs\expandafter
      {\@filef@und}{#2\@addtofilelist{#1}\@@input}}}
```

(*End definition for* `\InputIfFileExists`*. This function is documented on page* **??**.)

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

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩  {\@swaptwoargs}{Don't lose the file name}%
\long\def\@swaptwoargs#1#2{#2#1}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\@swaptwoargs}{Don't lose the file name}%
⟨latexrelease⟩\let\@swaptwoargs\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@swaptwoargs`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\input`*. This function is documented on page* **??**.)

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

Changes to `\@iinput`: adapt to the changes to `\@missingfileerror`.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@iinput}{Change in file error handling}%
\def\@iinput#1{%
  \InputIfFileExists{#1}{}%
  {\filename@parse\@curr@file
   \edef\reserved@a{\noexpand\@missingfileerror
     {\filename@area\filename@base}%
     {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
```

This line now just sets \@missingfile@⟨*part*⟩:

```
    \reserved@a
```

Now here we have to use it. The file here is guaranteed to exist, because \@missingfileerror ensures so, but we have to use \InputIfFileExists because it executes the file hooks.

```
    \edef\reserved@a{\noexpand\@iinput{%
      \@missingfile@area\@missingfile@base.\@missingfile@ext}}%
    \reserved@a}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                 {\@iinput}{Quote file names}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@iinput#1{%
⟨latexrelease⟩  \InputIfFileExists{#1}{}%
⟨latexrelease⟩  {\filename@parse\@curr@file
⟨latexrelease⟩   \edef\reserved@a{\noexpand\@missingfileerror
⟨latexrelease⟩     {\filename@area\filename@base}%
⟨latexrelease⟩     {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
⟨latexrelease⟩   \reserved@a}}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@iinput}{Quote file names}%
⟨latexrelease⟩\def\@iinput#1{%
⟨latexrelease⟩  \InputIfFileExists{#1}{}%
⟨latexrelease⟩  {\filename@parse{#1}%
⟨latexrelease⟩   \edef\reserved@a{\noexpand\@missingfileerror
⟨latexrelease⟩     {\filename@area\filename@base}%
⟨latexrelease⟩     {\ifx\filename@ext\relax tex\else\filename@ext\fi}}%
⟨latexrelease⟩   \reserved@a}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@iinput. *This function is documented on page* ??.)

\@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.}}}
```

(*End definition for* \@input. *This function is documented on page* ??.)

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

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

(*End definition for* \@input@. *This function is documented on page* ??.)

\@missingfileerror This 'error' command avoids TEX's primitive missing file loop.

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

Changes to \@missingfileerror: rather than trying to input the file by force, now \@missingfileerror just returns three \@missingfile@⟨*part*⟩ and the caller macro is responsible for doing the right thing with it.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩        {\@missingfileerror}{Do not load missing file immediately}%
\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
```

If the user answers with ⟨*return*⟩, fallback to the `.tex` file (previously it did nothing).

```
      \let\@missingfile@area\@empty
      \let\@missingfile@base\@empty
      \def\@missingfile@ext{tex}%
    \else
```

Use `\batchmode\read-1 to` ⟨*tl*⟩ to end the TEX run, same as expl3 does (it was `\batchmode\@@end` before).

```
      \def\reserved@b{\batchmode\read-1 to \reserved@a}%
      \def\reserved@a{x}\ifx\reserved@a\@gtempa\reserved@b\fi
      \def\reserved@a{X}\ifx\reserved@a\@gtempa\reserved@b\fi
      \filename@parse\@gtempa
      \edef\filename@ext{%
        \ifx\filename@ext\relax#2\else\filename@ext\fi}%
     \edef\reserved@a{%
```

Only check `\IfFileExists` (it was `\InputIfFileExists`).

```
       \noexpand\IfFileExists
         {\filename@area\filename@base.\filename@ext}%
```

If the file exists, define `\@missingfile@`⟨*part*⟩.

```
         {\def\noexpand\@missingfile@area{\filename@area}%
          \def\noexpand\@missingfile@base{\filename@base}%
          \def\noexpand\@missingfile@ext {\filename@ext}}%
         {\noexpand\@missingfileerror
            {\filename@area\filename@base}{\filename@ext}}}%
      \reserved@a
    \fi
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩        {\@missingfileerror}{Do not load missing file immediately}%
⟨latexrelease⟩
⟨latexrelease⟩\gdef\@missingfileerror#1#2{%
⟨latexrelease⟩     \typeout{^^J! LaTeX Error: File ‘#1.#2’ not found.^^J^^J%
⟨latexrelease⟩      Type X to quit or <RETURN> to proceed,^^J%
⟨latexrelease⟩      or enter new name. (Default extension: #2)^^J}%
⟨latexrelease⟩     \message{Enter file name: }%
⟨latexrelease⟩      {\endlinechar\m@ne
⟨latexrelease⟩       \global\read\m@ne to\@gtempa}%
⟨latexrelease⟩    \ifx\@gtempa\@empty
```

```
⟨latexrelease⟩    \else
⟨latexrelease⟩      \def\reserved@a{x}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
⟨latexrelease⟩      \def\reserved@a{X}\ifx\reserved@a\@gtempa\batchmode\@@end\fi
⟨latexrelease⟩      \filename@parse\@gtempa
⟨latexrelease⟩      \edef\filename@ext{%
⟨latexrelease⟩        \ifx\filename@ext\relax#2\else\filename@ext\fi}%
⟨latexrelease⟩     \edef\reserved@a{%
⟨latexrelease⟩       \noexpand\InputIfFileExists
⟨latexrelease⟩         {\filename@area\filename@base.\filename@ext}%
⟨latexrelease⟩         {}%
⟨latexrelease⟩         {\noexpand\@missingfileerror
⟨latexrelease⟩            {\filename@area\filename@base}{\filename@ext}}}%
⟨latexrelease⟩      \reserved@a
⟨latexrelease⟩    \fi}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@missingfileerror`*. This function is documented on page* **??**.)

`\@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
```

1.2 Listing files

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

`\@filelist`

```
\let\@filelist\@gobble
```

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.

`\@addtofilelist`

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

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

`\listfiles`

```
\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⟩

(*End definition for* `\@obsoletefile` *and others. These functions are documented on page* ??.)

File o
ltoutenc.dtx

1 Font encodings

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

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

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

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

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

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

The syntax for declaring encoding-specific commands is:

`\DeclareTextCommand{`⟨*command*⟩`}{`⟨*encoding*⟩`}`
`[`⟨*number*⟩`][`⟨*default*⟩`]{`⟨*commands*⟩`}`

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

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

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

`\ProvideTextCommand{`⟨*command*⟩`}{`⟨*encoding*⟩`}`
`[`⟨*number*⟩`][`⟨*default*⟩`]{`⟨*commands*⟩`}`

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

`\DeclareTextSymbol{`⟨*command*⟩`}{`⟨*encoding*⟩`}{`⟨*slot*⟩`}`

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

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

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

`\DeclareTextAccent{`⟨*command*⟩`}{`⟨*encoding*⟩`}{`⟨*slot*⟩`}`

This command declares a text accent. The commands:

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

have the same effect.

\DeclareTextComposite{⟨*command*⟩}
{⟨*encoding*⟩}{⟨*argument*⟩}{⟨*slot*⟩}

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

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

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

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

\DeclareTextCompositeCommand{⟨*command*⟩}
{⟨*encoding*⟩}{⟨*argument*⟩}{⟨*text*⟩}

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

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

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

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

\UseTextSymbol{⟨*encoding*⟩}{⟨*command*⟩}

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

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

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

\UseTextAccent{⟨*encoding*⟩}{⟨*command*⟩}{⟨*text*⟩}

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

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

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

`\DeclareTextCommandDefault{`⟨*command*⟩`}{`⟨*definition*⟩`}`

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

```
\DeclareTextCommandDefault{\textonequarter}{\ensuremath {\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`.

1.1 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}
```

1.2 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.

1.3 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.
TU	generates tuenc.def for Unicode font encoding.
package	generates fontenc.sty for selecting encodings.
2ekernel	for the kernel commands.

1.4 Definitions for the kernel

1.4.1 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⟩
\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}
```

```
\gdef\TextSymbolUnavailable#1{%
   \@latex@error{%
      Command \protect#1 unavailable in encoding \cf@encoding%
   }\@eha}
```

The command \@inmathwarn produces a warning message if we are currently in math mode. Note that since this command is used inside text commands, it can't call \relax before the \ifmmode. This means that it is possible for the warning to fail to be issued at the beginning of a row of an halign whose template enters math mode. This is probably a bad feature, but there's not much that can be done about it, since adding a \relax would break ligatures and kerning between text symbols.

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

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

(*End definition for* \DeclareTextCommand *and others. These functions are documented on page* ??.)

\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
```

(*End definition for* \DeclareTextCommandDefault *and* \ProvideTextCommandDefault. *These functions are documented on page* ??.)

\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
```

(*End definition for* \DeclareTextAccent. *This function is documented on page* ??.)

\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}%
```

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

```
\let\maybe@load@fontshape\relax
\let\ignorespaces\relax
\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
```

(*End definition for* \add@accent*. This function is documented on page* ??.)

\hmode@bgroup

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

(*End definition for* \hmode@bgroup*. This function is documented on page* ??.)

\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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017/04/15}{\DeclareTextCompositeCommand}
⟨latexrelease⟩                              {test for undeclared accent}%
⟨*2ekernel | latexrelease⟩
\def\DeclareTextCompositeCommand#1#2#3#4{%
  \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
  \ifx\reserved@a\relax
   \DeclareTextCommand#1{#2}{%
     \@latex@error{\string#1 undeclared in encoding #2}\@eha}%
   \@latex@info{Composite with undeclared \string#1 in encoding #2}%
   \expandafter\let\expandafter\reserved@a\csname#2\string#1\endcsname
  \fi
  \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\@empty\endcsname{#4}%
  }
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\DeclareTextCompositeCommand}
⟨latexrelease⟩                              {test for undeclared accent}%
⟨latexrelease⟩\def\DeclareTextCompositeCommand#1#2#3#4{%
⟨latexrelease⟩  \expandafter\let\expandafter\reserved@a
⟨latexrelease⟩                       \csname#2\string#1\endcsname
⟨latexrelease⟩  \expandafter\expandafter\expandafter\ifx
⟨latexrelease⟩  \expandafter\@car\reserved@a\relax\relax\@nil
⟨latexrelease⟩                                \@text@composite \else
⟨latexrelease⟩      \edef\reserved@b##1{%
⟨latexrelease⟩         \def\expandafter\noexpand
⟨latexrelease⟩           \csname#2\string#1\endcsname####1{%
⟨latexrelease⟩           \noexpand\@text@composite
⟨latexrelease⟩             \expandafter\noexpand\csname#2\string#1\endcsname
⟨latexrelease⟩             ####1\noexpand\@empty\noexpand\@text@composite
⟨latexrelease⟩             {##1}}}%
⟨latexrelease⟩      \expandafter\reserved@b\expandafter{\reserved@a{##1}}%
⟨latexrelease⟩   \fi
⟨latexrelease⟩   \expandafter\def\csname\expandafter\string\csname
⟨latexrelease⟩      #2\endcsname\string#1-\string#3\@empty\endcsname{#4}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@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
```

(*End definition for* `\DeclareTextCompositeCommand` *and others. These functions are documented on page* ??.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                    {\UseTextAccent}{Make commands robust}%
```

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

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

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

```
\DeclareRobustCommand*\UseTextAccent[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}%
   }}
```

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

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\UseTextAccent}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\UseTextAccent
⟨latexrelease⟩\kernel@make@fragile\UseTextSymbol
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

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

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

(*End definition for* \UseTextAccent*,* \UseTextSymbol*, and* \@use@text@encoding*. These functions are documented on page* ??.)

\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
```

(*End definition for* \hmode@start@before@group*. This function is documented on page* ??.)

\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
```

(*End definition for* `\DeclareTextSymbolDefault` *and* `\DeclareTextAccentDefault`*. These functions are documented on page* ??.)

\UndeclareTextCommand

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

```
\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
```

(*End definition for* `\UndeclareTextCommand`*. This function is documented on page* ??.)

1.4.2 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
%}
```

(*End definition for* \patterns *and others. These functions are documented on page* ??.)

1.4.3 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
```

(*End definition for* \a. *This function is documented on page* ??.)

1.4.4 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{\textunderscore}{%
  \leavevmode \kern.06em\vbox{\hrule\@width.3em}}
```

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

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

```
\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}}
```

1.4.5 Math material

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

```
\DeclareRobustCommand{\$}{\ifmmode\mathdollar\else\textdollar\fi}
```

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

```
\protected\def\{{\ifmmode\lbrace\else\textbraceleft\fi}
\protected\def\}{\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
```

Default definition of the commabelow accent.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/10/01}{\textcommabelow}{comma accent}%
⟨*2ekernel | latexrelease⟩
\DeclareTextCommandDefault\textcommabelow[1]
  {\hmode@bgroup\ooalign{\null#1\crcr\hidewidth\raise-.31ex
   \hbox{\check@mathfonts\fontsize\ssf@size\z@
   \math@fontsfalse\selectfont,}\hidewidth}\egroup}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\textcommabelow}{comma accent}%
⟨latexrelease⟩\let\textcommabelow\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-G\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-K\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-k\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-L\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-l\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-N\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-n\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩  \let\csname\string\T1\string\c-R\endcsname\@undefined
```

```
⟨latexrelease⟩\expandafter
⟨latexrelease⟩   \let\csname\string\T1\string\c-r\endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

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

```
⟨latexrelease⟩\IncludeInRelease{2016/02/01}{\textcommaabove}{comma above}%
⟨*2ekernel | latexrelease⟩
\DeclareTextCommandDefault\textcommaabove[1]{%
  \hmode@bgroup
  \ooalign{%
    \hidewidth
    \raise.7ex\hbox{%
      \check@mathfonts\fontsize\ssf@size\z@\math@fontsfalse\selectfont‘%
    }%
   \hidewidth\crcr
   \null#1\crcr
  }%
  \egroup
}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\textcommaabove}{comma above}%
⟨latexrelease⟩\let\textcommaabove\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩   \let\csname\string\OT1\string\c-g\endcsname\@undefined
⟨latexrelease⟩\expandafter
⟨latexrelease⟩   \let\csname\string\T1\string\c-g\endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

1.5 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}
```

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

```
\ifx\textcommaabove\@undefined\else
\DeclareTextCompositeCommand{\c}{OT1}{g}{\textcommaabove{g}}
\fi
```

```
⟨/OT1⟩
```

1.6 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?
```

For Maltese, `\Hwithstroke` and `\hwithstroke` are needed.

```
\DeclareTextCommand{\Hwithstroke}{T1}
   {%
    \hmode@bgroup
    \vphantom{H}%
    \sbox\z@{H}%
    \ooalign{%
      H\cr
      \hidewidth
      \vrule
        height \dimexpr 0.7\ht\z@+0.1ex\relax
        depth  -0.7\ht\z@
        width  0.8\wd\z@
      \hidewidth\cr
    }%
    \egroup
   }
\DeclareTextCommand{\hwithstroke}{T1}
   {%
    \hmode@bgroup
    \vphantom{h}%
    \sbox\z@{h}%
    \ooalign{%
      h\cr
```

```
      \kern0.075\wd\z@
      \vrule
        height \dimexpr 0.7\ht\z@+0.1ex\relax
        depth  -0.7\ht\z@
        width  0.4\wd\z@
      \hidewidth\cr
    }%
    \egroup
  }
```

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{\guillemetleft}{T1}{19}
\DeclareTextSymbol{\guillemetright}{T1}{20}
% old Adobe names
\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}}
```

```
\ifx\textcommaabove\@undefined\else
\DeclareTextCompositeCommand{\c}{T1}{g}{\textcommaabove{g}}
\fi
```

```
\ifx\textcommabelow\@undefined\else
\DeclareTextCompositeCommand{\c}{T1}{G}{\textcommabelow{G}}
\DeclareTextCompositeCommand{\c}{T1}{K}{\textcommabelow{K}}
\DeclareTextCompositeCommand{\c}{T1}{k}{\textcommabelow{k}}
\DeclareTextCompositeCommand{\c}{T1}{L}{\textcommabelow{L}}
\DeclareTextCompositeCommand{\c}{T1}{l}{\textcommabelow{l}}
\DeclareTextCompositeCommand{\c}{T1}{N}{\textcommabelow{N}}
\DeclareTextCompositeCommand{\c}{T1}{n}{\textcommabelow{n}}
\DeclareTextCompositeCommand{\c}{T1}{R}{\textcommabelow{R}}
\DeclareTextCompositeCommand{\c}{T1}{r}{\textcommabelow{r}}
\fi
```

⟨/T1⟩

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

```
\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⟩

1.8 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⟩
```

1.9 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{\guillemetleft}{OT4}{174}
\DeclareTextSymbol{\guillemetright}{OT4}{175}
% old Adobe names
\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⟩
```

1.10 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⟩
```

1.11 Definitions for the TU encoding

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

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

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

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

```
⟨*TU⟩
```

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

```
\providecommand\UnicodeEncodingName{TU}
```

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

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname XeTeXrevision\endcsname\relax
```

```
 \begingroup\expandafter\expandafter\expandafter\endgroup
 \expandafter\ifx\csname directlua\endcsname\relax
```

Not LuaTeX or XeTeX, abort with a warning.

```
   \PackageWarningNoLine{fontenc}
     {\UnicodeEncodingName\space
      encoding is only available with XeTeX and LuaTeX.\MessageBreak
      Defaulting to T1 encoding}
     \def\encodingdefault{T1}
   \expandafter\expandafter\expandafter\endinput
```

```
 \else
```

LuaTeX. For LuaTEX 1.10+, define a Lua function to disable any handing by the font code. Otherwise we reload the font without TeX ligatures.

```
   \def\UnicodeFontTeXLigatures{+tlig;}
   \ifnum\luatexversion<110
```

```
     \def\reserved@a#1{%
       \def\@remove@tlig@##1{\@remove@tlig@##1\@nil#1\@nil\relax}
       \def\@remove@tlig@##1#1{\@remove@tlig@@##1}}
     \edef\reserved@b{\detokenize{+tlig;}}
     \expandafter\reserved@a\expandafter{\reserved@b}
     \def\@remove@tlig@@#1\@nil#2\relax{#1}
```

```
     \def\remove@tlig#1{%
       \begingroup
       \font\remove@tlig
       \expandafter\@remove@tlig\expandafter{\fontname\font}%
       \remove@tlig
       \char#1\relax
       \endgroup
     }
   \else
     \newluafunction\@remove@tlig@@@@
```

Now we can define the function. Mostly we just have to insert a protected glyph node, which is a glyph node with subtype 256. But we have to keep track of the current mode to avoid inserting the glyph into a vlist.

```
     \now@and@everyjob{\directlua{
       local rawchar_func = token.create'@remove@tlig@@@@'.index
       local forcehmode = tex.forcehmode
       local put_next = token.put_next
       local glyph_id = node.id'glyph'
       local rawchar_token = token.new(rawchar_func, token.command_id'lua_call')
       lua.get_functions_table()[rawchar_func] = function()
         local mode = tex.nest.top.mode
         if mode == 1 or mode == -1 then
           put_next(rawchar_token)
           return forcehmode(true)
```

```
          end
          local n = node.new(glyph_id, 256)
          n.font = font.current()
          n.char = token.scan_int()
          return node.write(n)
        end
        token.set_lua('@remove@tlig@@@', rawchar_func, 'global', 'protected')
      }}
```

Now `\remove@tlig` can be implemented almost as in XeTeX.

```
      \def\remove@tlig#1{\@remove@tlig@@@#1\relax}
    \fi
  \fi
\else
```

XeTeX

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

Declare the encoding

```
\DeclareFontEncoding\UnicodeEncodingName{}{}
```

Declare accent command to use a postpended combining character rather than the TeX `\accent` primitive

```
\def\add@unicode@accent#1#2{%
  \if\relax\detokenize{#2}\relax^^a0\else#2\fi
  \char#1\relax}
\def\DeclareUnicodeAccent#1#2#3{%
  \DeclareTextCommand{#1}{#2}{\add@unicode@accent{#3}}%
}
```

Wrapper around `\DeclareTextCompositeCommand` that uses the declared composite if it exists in the current font or falls back to the default definition for the TU accent if not.

```
{
\catcode\z@=11\relax
\gdef\DeclareUnicodeComposite#1#2#3{%
   \def\reserved@a##1##2{%
     \DeclareTextCompositeCommand#1\UnicodeEncodingName{#2}{%
   \iffontchar\font#3 ##2%
      \else ##1\fi}}%
    \expandafter\expandafter\expandafter\extract@default@composite
    \csname\UnicodeEncodingName\string#1\endcsname{#2}\@nil
   \bgroup
      \lccode\z@#3 %
      \lowercase{\egroup
      \expandafter\reserved@a\expandafter{\reserved@b}{^^@}}}%
}
```

```
\def\extract@default@composite#1{%
 \ifx\@text@composite#1%
   \expandafter\extract@default@composite@a
 \else
   \expandafter\extract@default@composite@b\expandafter#1%
 \fi}

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

\DeclareTextCommand\textquotesingle \UnicodeEncodingName{%
                                          \remove@tlig{"0027}}
\DeclareTextCommand\textasciigrave  \UnicodeEncodingName{%
                                          \remove@tlig{"0060}}
\DeclareTextCommand\textquotedbl    \UnicodeEncodingName{%
                                          \remove@tlig{"0022}}

\DeclareTextSymbol{\textdollar}          \UnicodeEncodingName{"0024}
\DeclareTextSymbol{\textless}            \UnicodeEncodingName{"003C}
\DeclareTextSymbol{\textgreater}         \UnicodeEncodingName{"003E}
\DeclareTextSymbol{\textbackslash}       \UnicodeEncodingName{"005C}
\DeclareTextSymbol{\textasciicircum}     \UnicodeEncodingName{"005E}
\DeclareTextSymbol{\textunderscore}      \UnicodeEncodingName{"005F}
\DeclareTextSymbol{\textbraceleft}       \UnicodeEncodingName{"007B}
\DeclareTextSymbol{\textbar}             \UnicodeEncodingName{"007C}
\DeclareTextSymbol{\textbraceright}      \UnicodeEncodingName{"007D}
\DeclareTextSymbol{\textasciitilde}      \UnicodeEncodingName{"007E}
\DeclareTextSymbol{\textexclamdown}      \UnicodeEncodingName{"00A1}
\DeclareTextSymbol{\textcent}            \UnicodeEncodingName{"00A2}
\DeclareTextSymbol{\textsterling}        \UnicodeEncodingName{"00A3}
\DeclareTextSymbol{\textcurrency}        \UnicodeEncodingName{"00A4}
\DeclareTextSymbol{\textyen}             \UnicodeEncodingName{"00A5}
\DeclareTextSymbol{\textbrokenbar}       \UnicodeEncodingName{"00A6}
\DeclareTextSymbol{\textsection}         \UnicodeEncodingName{"00A7}
\DeclareTextSymbol{\textasciidieresis}   \UnicodeEncodingName{"00A8}
\DeclareTextSymbol{\textcopyright}       \UnicodeEncodingName{"00A9}
\DeclareTextSymbol{\textordfeminine}     \UnicodeEncodingName{"00AA}

\DeclareTextSymbol{\guillemetleft}       \UnicodeEncodingName{"00AB}
% old Adobe name
\DeclareTextSymbol{\guillemotleft}       \UnicodeEncodingName{"00AB}

\DeclareTextSymbol{\textlnot}            \UnicodeEncodingName{"00AC}
\DeclareTextSymbol{\textregistered}      \UnicodeEncodingName{"00AE}
\DeclareTextSymbol{\textasciimacron}     \UnicodeEncodingName{"00AF}
\DeclareTextSymbol{\textdegree}          \UnicodeEncodingName{"00B0}
\DeclareTextSymbol{\textpm}              \UnicodeEncodingName{"00B1}
\DeclareTextSymbol{\texttwosuperior}     \UnicodeEncodingName{"00B2}
\DeclareTextSymbol{\textthreesuperior}   \UnicodeEncodingName{"00B3}
\DeclareTextSymbol{\textasciiacute}      \UnicodeEncodingName{"00B4}
\DeclareTextSymbol{\textmu}              \UnicodeEncodingName{"00B5}
\DeclareTextSymbol{\textparagraph}       \UnicodeEncodingName{"00B6}
\DeclareTextSymbol{\textperiodcentered}  \UnicodeEncodingName{"00B7}
\DeclareTextSymbol{\textonesuperior}     \UnicodeEncodingName{"00B9}
\DeclareTextSymbol{\textordmasculine}    \UnicodeEncodingName{"00BA}
```

```
\DeclareTextSymbol{\guillemetright}      \UnicodeEncodingName{"00BB}
% old Adobe name
\DeclareTextSymbol{\guillemotright}      \UnicodeEncodingName{"00BB}
\DeclareTextSymbol{\textonequarter}      \UnicodeEncodingName{"00BC}
\DeclareTextSymbol{\textonehalf}         \UnicodeEncodingName{"00BD}
\DeclareTextSymbol{\textthreequarters}   \UnicodeEncodingName{"00BE}
\DeclareTextSymbol{\textquestiondown}    \UnicodeEncodingName{"00BF}
\DeclareTextSymbol{\AE}                  \UnicodeEncodingName{"00C6}
\DeclareTextSymbol{\DH}                  \UnicodeEncodingName{"00D0}
\DeclareTextSymbol{\texttimes}           \UnicodeEncodingName{"00D7}
\DeclareTextSymbol{\O}                   \UnicodeEncodingName{"00D8}
\DeclareTextSymbol{\TH}                  \UnicodeEncodingName{"00DE}
\DeclareTextSymbol{\ss}                  \UnicodeEncodingName{"00DF}
\DeclareTextSymbol{\ae}                  \UnicodeEncodingName{"00E6}
\DeclareTextSymbol{\dh}                  \UnicodeEncodingName{"00F0}
\DeclareTextSymbol{\textdiv}             \UnicodeEncodingName{"00F7}
\DeclareTextSymbol{\o}                   \UnicodeEncodingName{"00F8}
\DeclareTextSymbol{\th}                  \UnicodeEncodingName{"00FE}
\DeclareTextSymbol{\DJ}                  \UnicodeEncodingName{"0110}
\DeclareTextSymbol{\dj}                  \UnicodeEncodingName{"0111}
\DeclareTextSymbol{\i}                   \UnicodeEncodingName{"0131}
\DeclareTextSymbol{\IJ}                  \UnicodeEncodingName{"0132}
\DeclareTextSymbol{\ij}                  \UnicodeEncodingName{"0133}
\DeclareTextSymbol{\L}                   \UnicodeEncodingName{"0141}
\DeclareTextSymbol{\l}                   \UnicodeEncodingName{"0142}
\DeclareTextSymbol{\NG}                  \UnicodeEncodingName{"014A}
\DeclareTextSymbol{\ng}                  \UnicodeEncodingName{"014B}
\DeclareTextSymbol{\OE}                  \UnicodeEncodingName{"0152}
\DeclareTextSymbol{\oe}                  \UnicodeEncodingName{"0153}
\DeclareTextSymbol{\textflorin}          \UnicodeEncodingName{"0192}
\DeclareTextSymbol{\j}                   \UnicodeEncodingName{"0237}
\DeclareTextSymbol{\textasciicaron}      \UnicodeEncodingName{"02C7}
\DeclareTextSymbol{\textasciibreve}      \UnicodeEncodingName{"02D8}
\DeclareTextSymbol{\textacutedbl}        \UnicodeEncodingName{"02DD}
\DeclareTextSymbol{\textgravedbl}        \UnicodeEncodingName{"02F5}
\DeclareTextSymbol{\texttildelow}        \UnicodeEncodingName{"02F7}
\DeclareTextSymbol{\textbaht}            \UnicodeEncodingName{"0E3F}
\DeclareTextSymbol{\SS}                  \UnicodeEncodingName{"1E9E}
\DeclareTextSymbol{\textcompwordmark}    \UnicodeEncodingName{"200C}
\DeclareTextSymbol{\textendash}          \UnicodeEncodingName{"2013}
\DeclareTextSymbol{\textemdash}          \UnicodeEncodingName{"2014}
\DeclareTextSymbol{\textbardbl}          \UnicodeEncodingName{"2016}
\DeclareTextSymbol{\textquoteleft}       \UnicodeEncodingName{"2018}
\DeclareTextSymbol{\textquoteright}      \UnicodeEncodingName{"2019}
\DeclareTextSymbol{\quotesinglbase}      \UnicodeEncodingName{"201A}
\DeclareTextSymbol{\textquotedblleft}    \UnicodeEncodingName{"201C}
\DeclareTextSymbol{\textquotedblright}   \UnicodeEncodingName{"201D}
\DeclareTextSymbol{\quotedblbase}        \UnicodeEncodingName{"201E}
\DeclareTextSymbol{\textdagger}          \UnicodeEncodingName{"2020}
\DeclareTextSymbol{\textdaggerdbl}       \UnicodeEncodingName{"2021}
\DeclareTextSymbol{\textbullet}          \UnicodeEncodingName{"2022}
\DeclareTextSymbol{\textellipsis}        \UnicodeEncodingName{"2026}
\DeclareTextSymbol{\textperthousand}     \UnicodeEncodingName{"2030}
\DeclareTextSymbol{\textpertenthousand}  \UnicodeEncodingName{"2031}
```

```
\DeclareTextSymbol{\guilsinglleft}       \UnicodeEncodingName{"2039}
\DeclareTextSymbol{\guilsinglright}      \UnicodeEncodingName{"203A}
\DeclareTextSymbol{\textreferencemark}   \UnicodeEncodingName{"203B}
\DeclareTextSymbol{\textinterrobang}     \UnicodeEncodingName{"203D}
\DeclareTextSymbol{\textfractionsolidus} \UnicodeEncodingName{"2044}
\DeclareTextSymbol{\textlquill}          \UnicodeEncodingName{"2045}
\DeclareTextSymbol{\textrquill}          \UnicodeEncodingName{"2046}
\DeclareTextSymbol{\textdiscount}        \UnicodeEncodingName{"2052}
\DeclareTextSymbol{\textcolonmonetary}   \UnicodeEncodingName{"20A1}
\DeclareTextSymbol{\textlira}            \UnicodeEncodingName{"20A4}
\DeclareTextSymbol{\textnaira}           \UnicodeEncodingName{"20A6}
\DeclareTextSymbol{\textwon}             \UnicodeEncodingName{"20A9}
\DeclareTextSymbol{\textdong}            \UnicodeEncodingName{"20AB}
\DeclareTextSymbol{\texteuro}            \UnicodeEncodingName{"20AC}
\DeclareTextSymbol{\textpeso}            \UnicodeEncodingName{"20B1}
\DeclareTextSymbol{\textcelsius}         \UnicodeEncodingName{"2103}
\DeclareTextSymbol{\textnumero}          \UnicodeEncodingName{"2116}
\DeclareTextSymbol{\textcircledP}        \UnicodeEncodingName{"2117}
\DeclareTextSymbol{\textrecipe}          \UnicodeEncodingName{"211E}
\DeclareTextSymbol{\textservicemark}     \UnicodeEncodingName{"2120}
\DeclareTextSymbol{\texttrademark}       \UnicodeEncodingName{"2122}
\DeclareTextSymbol{\textohm}             \UnicodeEncodingName{"2126}
\DeclareTextSymbol{\textmho}             \UnicodeEncodingName{"2127}
\DeclareTextSymbol{\textestimated}       \UnicodeEncodingName{"212E}
\DeclareTextSymbol{\textleftarrow}       \UnicodeEncodingName{"2190}
\DeclareTextSymbol{\textuparrow}         \UnicodeEncodingName{"2191}
\DeclareTextSymbol{\textrightarrow}      \UnicodeEncodingName{"2192}
\DeclareTextSymbol{\textdownarrow}       \UnicodeEncodingName{"2193}
\DeclareTextSymbol{\textminus}           \UnicodeEncodingName{"2212}

```

```
\DeclareTextSymbol{\Hwithstroke}         \UnicodeEncodingName{"0126}
\DeclareTextSymbol{\hwithstroke}         \UnicodeEncodingName{"0127}
```

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

```
\DeclareTextCommand{\textasteriskcentered}\UnicodeEncodingName{%
  \iffontchar\font"2217 \char"2217 \else
    \begingroup
      \fontsize
       {\the\dimexpr1.2\dimexpr\f@size pt\relax}%
       {\f@baselineskip}%
      \selectfont
      \raisebox{-0.6ex}[\dimexpr\height-0.6ex][0pt]{*}%
    \endgroup
  \fi
}
```

```
\DeclareTextSymbol{\textsurd}            \UnicodeEncodingName{"221A}
\DeclareTextSymbol{\textlangle}          \UnicodeEncodingName{"2329}
\DeclareTextSymbol{\textrangle}          \UnicodeEncodingName{"232A}
\DeclareTextSymbol{\textblank}           \UnicodeEncodingName{"2422}
\DeclareTextSymbol{\textvisiblespace}    \UnicodeEncodingName{"2423}
\DeclareTextSymbol{\textopenbullet}      \UnicodeEncodingName{"25E6}
\DeclareTextSymbol{\textbigcircle}       \UnicodeEncodingName{"25EF}
\DeclareTextSymbol{\textmusicalnote}     \UnicodeEncodingName{"266A}
```

```
\DeclareTextSymbol{\textmarried}         \UnicodeEncodingName{"26AD}
\DeclareTextSymbol{\textdivorced}        \UnicodeEncodingName{"26AE}
\DeclareTextSymbol{\textinterrobangdown} \UnicodeEncodingName{"2E18}
```

Accents must be declared before the composites that use them.

```
\DeclareUnicodeAccent{\`}                \UnicodeEncodingName{"0300}
\DeclareUnicodeAccent{\'}                \UnicodeEncodingName{"0301}
\DeclareUnicodeAccent{\^}                \UnicodeEncodingName{"0302}
\DeclareUnicodeAccent{\~}                \UnicodeEncodingName{"0303}
\DeclareUnicodeAccent{\=}                \UnicodeEncodingName{"0304}
\DeclareUnicodeAccent{\u}                \UnicodeEncodingName{"0306}
\DeclareUnicodeAccent{\.}                \UnicodeEncodingName{"0307}
\DeclareUnicodeAccent{\"}                \UnicodeEncodingName{"0308}
\DeclareUnicodeAccent{\r}                \UnicodeEncodingName{"030A}
\DeclareUnicodeAccent{\H}                \UnicodeEncodingName{"030B}
\DeclareUnicodeAccent{\v}                \UnicodeEncodingName{"030C}
\DeclareUnicodeAccent{\b}                \UnicodeEncodingName{"0332}
\DeclareUnicodeAccent{\d}                \UnicodeEncodingName{"0323}
\DeclareUnicodeAccent{\c}                \UnicodeEncodingName{"0327}
\DeclareUnicodeAccent{\k}                \UnicodeEncodingName{"0328}
\DeclareTextCommand\textcommabelow       \UnicodeEncodingName[1]
  {\hmode@bgroup\ooalign{\null#1\crcr\hidewidth\raise-.31ex
   \hbox{\check@mathfonts\fontsize\ssf@size\z@
   \math@fontsfalse\selectfont,}\hidewidth}\egroup}
```

```
\DeclareUnicodeComposite{\^}              {}{"005E}
\DeclareUnicodeComposite{\~}              {}{"007E}
```

```
\DeclareUnicodeComposite{\`}             {A}{"00C0}
\DeclareUnicodeComposite{\'}             {A}{"00C1}
\DeclareUnicodeComposite{\^}             {A}{"00C2}
\DeclareUnicodeComposite{\~}             {A}{"00C3}
\DeclareUnicodeComposite{\"}             {A}{"00C4}
\DeclareUnicodeComposite{\r}             {A}{"00C5}
\DeclareUnicodeComposite{\c}             {C}{"00C7}
\DeclareUnicodeComposite{\`}             {E}{"00C8}
\DeclareUnicodeComposite{\'}             {E}{"00C9}
\DeclareUnicodeComposite{\^}             {E}{"00CA}
\DeclareUnicodeComposite{\"}             {E}{"00CB}
\DeclareUnicodeComposite{\`}             {I}{"00CC}
\DeclareUnicodeComposite{\'}             {I}{"00CD}
\DeclareUnicodeComposite{\^}             {I}{"00CE}
\DeclareUnicodeComposite{\"}             {I}{"00CF}
\DeclareUnicodeComposite{\~}             {N}{"00D1}
\DeclareUnicodeComposite{\`}             {O}{"00D2}
\DeclareUnicodeComposite{\'}             {O}{"00D3}
\DeclareUnicodeComposite{\^}             {O}{"00D4}
\DeclareUnicodeComposite{\~}             {O}{"00D5}
\DeclareUnicodeComposite{\"}             {O}{"00D6}
\DeclareUnicodeComposite{\`}             {U}{"00D9}
\DeclareUnicodeComposite{\'}             {U}{"00DA}
\DeclareUnicodeComposite{\^}             {U}{"00DB}
\DeclareUnicodeComposite{\"}             {U}{"00DC}
\DeclareUnicodeComposite{\'}             {Y}{"00DD}
\DeclareUnicodeComposite{\`}             {a}{"00E0}
\DeclareUnicodeComposite{\'}             {a}{"00E1}
```

```
\DeclareUnicodeComposite{\^}            {a}{"00E2}
\DeclareUnicodeComposite{\~}            {a}{"00E3}
\DeclareUnicodeComposite{\"}            {a}{"00E4}
\DeclareUnicodeComposite{\r}            {a}{"00E5}
\DeclareUnicodeComposite{\c}            {c}{"00E7}
\DeclareUnicodeComposite{\`}            {e}{"00E8}
\DeclareUnicodeComposite{\'}            {e}{"00E9}
\DeclareUnicodeComposite{\^}            {e}{"00EA}
\DeclareUnicodeComposite{\"}            {e}{"00EB}
\DeclareUnicodeComposite{\`}            \i {"00EC}
\DeclareUnicodeComposite{\`}            {i}{"00EC}
\DeclareUnicodeComposite{\'}            \i {"00ED}
\DeclareUnicodeComposite{\'}            {i}{"00ED}
\DeclareUnicodeComposite{\^}            \i {"00EE}
\DeclareUnicodeComposite{\^}            {i}{"00EE}
\DeclareUnicodeComposite{\"}            \i {"00EF}
\DeclareUnicodeComposite{\"}            {i}{"00EF}
\DeclareUnicodeComposite{\~}            {n}{"00F1}
\DeclareUnicodeComposite{\`}            {o}{"00F2}
\DeclareUnicodeComposite{\'}            {o}{"00F3}
\DeclareUnicodeComposite{\^}            {o}{"00F4}
\DeclareUnicodeComposite{\~}            {o}{"00F5}
\DeclareUnicodeComposite{\"}            {o}{"00F6}
\DeclareUnicodeComposite{\`}            {u}{"00F9}
\DeclareUnicodeComposite{\'}            {u}{"00FA}
\DeclareUnicodeComposite{\^}            {u}{"00FB}
\DeclareUnicodeComposite{\"}            {u}{"00FC}
\DeclareUnicodeComposite{\'}            {y}{"00FD}
\DeclareUnicodeComposite{\"}            {y}{"00FF}
\DeclareUnicodeComposite{\=}            {A}{"0100}
\DeclareUnicodeComposite{\=}            {a}{"0101}
\DeclareUnicodeComposite{\u}            {A}{"0102}
\DeclareUnicodeComposite{\u}            {a}{"0103}
\DeclareUnicodeComposite{\k}            {A}{"0104}
\DeclareUnicodeComposite{\k}            {a}{"0105}
\DeclareUnicodeComposite{\'}            {C}{"0106}
\DeclareUnicodeComposite{\'}            {c}{"0107}
\DeclareUnicodeComposite{\^}            {C}{"0108}
\DeclareUnicodeComposite{\^}            {c}{"0109}
\DeclareUnicodeComposite{\.}            {C}{"010A}
\DeclareUnicodeComposite{\.}            {c}{"010B}
\DeclareUnicodeComposite{\v}            {C}{"010C}
\DeclareUnicodeComposite{\v}            {c}{"010D}
\DeclareUnicodeComposite{\v}            {D}{"010E}
\DeclareUnicodeComposite{\v}            {d}{"010F}
\DeclareUnicodeComposite{\=}            {E}{"0112}
\DeclareUnicodeComposite{\=}            {e}{"0113}
\DeclareUnicodeComposite{\u}            {E}{"0114}
\DeclareUnicodeComposite{\u}            {e}{"0115}
\DeclareUnicodeComposite{\.}            {E}{"0116}
\DeclareUnicodeComposite{\.}            {e}{"0117}
\DeclareUnicodeComposite{\k}            {E}{"0118}
\DeclareUnicodeComposite{\k}            {e}{"0119}
\DeclareUnicodeComposite{\v}            {E}{"011A}
```

```
\DeclareUnicodeComposite{\v}            {e}{"011B}
\DeclareUnicodeComposite{\^}            {G}{"011C}
\DeclareUnicodeComposite{\^}            {g}{"011D}
\DeclareUnicodeComposite{\u}            {G}{"011E}
\DeclareUnicodeComposite{\u}            {g}{"011F}
\DeclareUnicodeComposite{\.}            {G}{"0120}
\DeclareUnicodeComposite{\.}            {g}{"0121}
\DeclareUnicodeComposite{\c}            {G}{"0122}
\DeclareUnicodeComposite{\c}            {g}{"0123}
\DeclareUnicodeComposite{\^}            {H}{"0124}
\DeclareUnicodeComposite{\^}            {h}{"0125}
\DeclareUnicodeComposite{\~}            {I}{"0128}
\DeclareUnicodeComposite{\~}            \i {"0129}
\DeclareUnicodeComposite{\~}            {i}{"0129}
\DeclareUnicodeComposite{\=}            {I}{"012A}
\DeclareUnicodeComposite{\=}            \i {"012B}
\DeclareUnicodeComposite{\=}            {i}{"012B}
\DeclareUnicodeComposite{\u}            {I}{"012C}
\DeclareUnicodeComposite{\u}            \i {"012D}
\DeclareUnicodeComposite{\u}            {i}{"012D}
\DeclareUnicodeComposite{\k}            {I}{"012E}
\DeclareUnicodeComposite{\k}            \i {"012F}
\DeclareUnicodeComposite{\k}            {i}{"012F}
\DeclareUnicodeComposite{\.}            {I}{"0130}
\DeclareUnicodeComposite{\^}            {J}{"0134}
\DeclareUnicodeComposite{\^}            \j {"0135}
\DeclareUnicodeComposite{\^}            {j}{"0135}
\DeclareUnicodeComposite{\c}            {K}{"0136}
\DeclareUnicodeComposite{\c}            {k}{"0137}
\DeclareUnicodeComposite{\'}            {L}{"0139}
\DeclareUnicodeComposite{\'}            {l}{"013A}
\DeclareUnicodeComposite{\c}            {L}{"013B}
\DeclareUnicodeComposite{\c}            {l}{"013C}
\DeclareUnicodeComposite{\v}            {L}{"013D}
\DeclareUnicodeComposite{\v}            {l}{"013E}
\DeclareUnicodeComposite{\'}            {N}{"0143}
\DeclareUnicodeComposite{\'}            {n}{"0144}
\DeclareUnicodeComposite{\c}            {N}{"0145}
\DeclareUnicodeComposite{\c}            {n}{"0146}
\DeclareUnicodeComposite{\v}            {N}{"0147}
\DeclareUnicodeComposite{\v}            {n}{"0148}
\DeclareUnicodeComposite{\=}            {O}{"014C}
\DeclareUnicodeComposite{\=}            {o}{"014D}
\DeclareUnicodeComposite{\u}            {O}{"014E}
\DeclareUnicodeComposite{\u}            {o}{"014F}
\DeclareUnicodeComposite{\H}            {O}{"0150}
\DeclareUnicodeComposite{\H}            {o}{"0151}
\DeclareUnicodeComposite{\'}            {R}{"0154}
\DeclareUnicodeComposite{\'}            {r}{"0155}
\DeclareUnicodeComposite{\c}            {R}{"0156}
\DeclareUnicodeComposite{\c}            {r}{"0157}
\DeclareUnicodeComposite{\v}            {R}{"0158}
\DeclareUnicodeComposite{\v}            {r}{"0159}
\DeclareUnicodeComposite{\'}            {S}{"015A}
```

```
\DeclareUnicodeComposite{\'}            {s}{"015B}
\DeclareUnicodeComposite{\^}            {S}{"015C}
\DeclareUnicodeComposite{\^}            {s}{"015D}
\DeclareUnicodeComposite{\c}            {S}{"015E}
\DeclareUnicodeComposite{\c}            {s}{"015F}
\DeclareUnicodeComposite{\v}            {S}{"0160}
\DeclareUnicodeComposite{\v}            {s}{"0161}
\DeclareUnicodeComposite{\c}            {T}{"0162}
\DeclareUnicodeComposite{\c}            {t}{"0163}
\DeclareUnicodeComposite{\v}            {T}{"0164}
\DeclareUnicodeComposite{\v}            {t}{"0165}
\DeclareUnicodeComposite{\~}            {U}{"0168}
\DeclareUnicodeComposite{\~}            {u}{"0169}
\DeclareUnicodeComposite{\=}            {U}{"016A}
\DeclareUnicodeComposite{\=}            {u}{"016B}
\DeclareUnicodeComposite{\u}            {U}{"016C}
\DeclareUnicodeComposite{\u}            {u}{"016D}
\DeclareUnicodeComposite{\r}            {U}{"016E}
\DeclareUnicodeComposite{\r}            {u}{"016F}
\DeclareUnicodeComposite{\H}            {U}{"0170}
\DeclareUnicodeComposite{\H}            {u}{"0171}
\DeclareUnicodeComposite{\k}            {U}{"0172}
\DeclareUnicodeComposite{\k}            {u}{"0173}
\DeclareUnicodeComposite{\^}            {W}{"0174}
\DeclareUnicodeComposite{\^}            {w}{"0175}
\DeclareUnicodeComposite{\^}            {Y}{"0176}
\DeclareUnicodeComposite{\^}            {y}{"0177}
\DeclareUnicodeComposite{\"}            {Y}{"0178}
\DeclareUnicodeComposite{\'}            {Z}{"0179}
\DeclareUnicodeComposite{\'}            {z}{"017A}
\DeclareUnicodeComposite{\.}            {Z}{"017B}
\DeclareUnicodeComposite{\.}            {z}{"017C}
\DeclareUnicodeComposite{\v}            {Z}{"017D}
\DeclareUnicodeComposite{\v}            {z}{"017E}
\DeclareUnicodeComposite{\v}            {A}{"01CD}
\DeclareUnicodeComposite{\v}            {a}{"01CE}
\DeclareUnicodeComposite{\v}            {I}{"01CF}
\DeclareUnicodeComposite{\v}            \i {"01D0}
\DeclareUnicodeComposite{\v}            {i}{"01D0}
\DeclareUnicodeComposite{\v}            {O}{"01D1}
\DeclareUnicodeComposite{\v}            {o}{"01D2}
\DeclareUnicodeComposite{\v}            {U}{"01D3}
\DeclareUnicodeComposite{\v}            {u}{"01D4}
\DeclareUnicodeComposite{\=}            \AE{"01E2}
\DeclareUnicodeComposite{\=}            \ae{"01E3}
\DeclareUnicodeComposite{\v}            {G}{"01E6}
\DeclareUnicodeComposite{\v}            {g}{"01E7}
\DeclareUnicodeComposite{\v}            {K}{"01E8}
\DeclareUnicodeComposite{\v}            {k}{"01E9}
\DeclareUnicodeComposite{\k}            {O}{"01EA}
\DeclareUnicodeComposite{\k}            {o}{"01EB}
\DeclareUnicodeComposite{\v}            \j {"01F0}
\DeclareUnicodeComposite{\v}            {j}{"01F0}
\DeclareUnicodeComposite{\'}            {G}{"01F4}
```

```
\DeclareUnicodeComposite{\'}             {g}{"01F5}
\DeclareUnicodeComposite{\textcommabelow}{S}{"0218}
\DeclareUnicodeComposite{\textcommabelow}{s}{"0219}
\DeclareUnicodeComposite{\textcommabelow}{T}{"021A}
\DeclareUnicodeComposite{\textcommabelow}{t}{"021B}
\DeclareUnicodeComposite{\=}             {Y}{"0232}
\DeclareUnicodeComposite{\=}             {y}{"0233}
\DeclareUnicodeComposite{\.}             {B}{"1E02}
\DeclareUnicodeComposite{\.}             {b}{"1E03}
\DeclareUnicodeComposite{\d}             {B}{"1E04}
\DeclareUnicodeComposite{\d}             {b}{"1E05}
\DeclareUnicodeComposite{\d}             {D}{"1E0C}
\DeclareUnicodeComposite{\d}             {d}{"1E0D}
\DeclareUnicodeComposite{\=}             {G}{"1E20}
\DeclareUnicodeComposite{\=}             {g}{"1E21}
\DeclareUnicodeComposite{\d}             {H}{"1E24}
\DeclareUnicodeComposite{\d}             {h}{"1E25}
\DeclareUnicodeComposite{\d}             {K}{"1E32}
\DeclareUnicodeComposite{\d}             {k}{"1E33}
\DeclareUnicodeComposite{\d}             {L}{"1E36}
\DeclareUnicodeComposite{\d}             {l}{"1E37}
\DeclareUnicodeComposite{\d}             {M}{"1E42}
\DeclareUnicodeComposite{\d}             {m}{"1E43}
\DeclareUnicodeComposite{\d}             {N}{"1E46}
\DeclareUnicodeComposite{\d}             {n}{"1E47}
\DeclareUnicodeComposite{\d}             {R}{"1E5A}
\DeclareUnicodeComposite{\d}             {r}{"1E5B}
\DeclareUnicodeComposite{\d}             {S}{"1E62}
\DeclareUnicodeComposite{\d}             {s}{"1E63}
\DeclareUnicodeComposite{\d}             {T}{"1E6C}
\DeclareUnicodeComposite{\d}             {t}{"1E6D}
\DeclareUnicodeComposite{\d}             {V}{"1E7E}
\DeclareUnicodeComposite{\d}             {v}{"1E7F}
\DeclareUnicodeComposite{\d}             {W}{"1E88}
\DeclareUnicodeComposite{\d}             {w}{"1E89}
\DeclareUnicodeComposite{\d}             {Z}{"1E92}
\DeclareUnicodeComposite{\d}             {z}{"1E93}
\DeclareUnicodeComposite{\d}             {A}{"1EA0}
\DeclareUnicodeComposite{\d}             {a}{"1EA1}
\DeclareUnicodeComposite{\d}             {E}{"1EB8}
\DeclareUnicodeComposite{\d}             {e}{"1EB9}
\DeclareUnicodeComposite{\d}             {I}{"1ECA}
\DeclareUnicodeComposite{\d}             {i}{"1ECB}
\DeclareUnicodeComposite{\d}             {O}{"1ECC}
\DeclareUnicodeComposite{\d}             {o}{"1ECD}
\DeclareUnicodeComposite{\d}             {U}{"1EE4}
\DeclareUnicodeComposite{\d}             {u}{"1EE5}
\DeclareUnicodeComposite{\d}             {Y}{"1EF4}
\DeclareUnicodeComposite{\d}             {y}{"1EF5}
```

⟨/TU⟩

2 Package files

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

2.1 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 afterwards 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
```

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

```
  \@tempswafalse
  \@ifl@t@r\fmtversion{2020/02/02}%
      {\expandafter\ifx\csname T@\CurrentOption\endcsname\relax
        \@tempswatrue\fi}%
      {\@tempswatrue}%
```

Load if necessary:

```
  \if@tempswa
    \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
  \fi
}
```

```
\ProcessOptions*
```

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

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

So instead we do this here:

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

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.

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

File p
ltcounts.dtx

1 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*⟩`}`: ∗, †, ‡, ...

`\counterwithin` `\counterwithin{`⟨*counter*⟩`}{`⟨*within-counter*⟩`}`: Resets ⟨*counter*⟩ whenever ⟨*within-counter*⟩ is stepped. Also redefines `\the`⟨*counter*⟩ command to produce `\the`⟨*within-counter*⟩`.\arabic{`⟨*counter*⟩`}` Star form omits redefining the print representation.

`\counterwithout` `\counterwithout{`⟨*counter*⟩`}{`⟨*within-counter*⟩`}`: Removes ⟨*counter*⟩ from the reset list of ⟨*within-counter*⟩. Also redefines `\the`⟨*counter*⟩ command to produce `\arabic{`⟨*counter*⟩`}`. Star form omits redefining the print representation.

```
⟨*2ekernel⟩
```

1.1 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.

\@removefromreset{⟨*foo*⟩}{⟨*bar*⟩} : Removes counter ⟨*foo*⟩ to the list of counters \cl@bar to be reset when counter ⟨*bar*⟩ is stepped.

\setcounter

\setcounter{⟨*foo*⟩}{⟨*val*⟩} : Globally sets \foocounter equal to ⟨*val*⟩.

```
\def\setcounter#1#2{%
  \@ifundefined{c@#1}%
    {\@nocounterr{#1}}%
    {\global\csname c@#1\endcsname#2\relax}}
```

(*End definition for* \setcounter*. This function is documented on page* ??.)

\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}}
```

(*End definition for* \addtocounter*. This function is documented on page* ??.)

\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}}{}}
```

(*End definition for* \newcounter*. This function is documented on page* ??.)

\value

\value{⟨*ctr*⟩} produces the value of counter ⟨*ctr*⟩, for use with a \setcounter or \addtocounter command.

```
\def\value#1{\csname c@#1\endcsname}
```

(*End definition for* \value*. This function is documented on page* ??.)

\@newctr

```
\def\@newctr#1[#2]{%
  \@ifundefined{c@#2}{\@nocounterr{#2}}{\@addtoreset{#1}{#2}}}
```

(*End definition for* \@newctr*. This function is documented on page* ??.)

\stepcounter

\stepcounterfoo 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}
```

(*End definition for* \stepcounter*. This function is documented on page* ??.)

\@stpelt

Rather than resetting the "within" counter to zero we set it to −1 and then run \stepcounter that moves it to 0 and also initiates resetting the next level down.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@stpelt}
⟨latexrelease⟩                              {Reset nested counters}%
⟨*2ekernel | latexrelease⟩
\def\@stpelt#1{\global\csname c@#1\endcsname \m@ne\stepcounter{#1}}%
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@stpelt}
⟨latexrelease⟩                              {Reset nested counters}%%
⟨latexrelease⟩\def\@stpelt#1{\global\csname c@#1\endcsname \z@}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@stpelt*. This function is documented on page* ??.)

\cl@@ckpt

```
\def\cl@@ckpt{\@elt{page}}
```

(*End definition for* \cl@@ckpt*. This function is documented on 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}}
```

(*End definition for* \@definecounter*. This function is documented on page* ??.)

\@addtoreset

```
\def\@addtoreset#1#2{\expandafter\@cons\csname cl@#2\endcsname {{#1}}}
```

(*End definition for* \@addtoreset*. This function is documented on page* ??.)

```
⟨/2ekernel⟩
```

\@removefromreset

```
⟨latexrelease⟩\IncludeInRelease{2018-04-01}
⟨latexrelease⟩                 {\@removefromreset}{Add interfaces}%
⟨*2ekernel | latexrelease⟩
```

```
\def\@removefromreset#1#2{%
```

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

```
  \@ifundefined{c@#2}\relax
  {\begingroup
     \expandafter\let\csname c@#1\endcsname\@removefromreset
     \def\@elt##1{%
       \expandafter\ifx\csname c@##1\endcsname\@removefromreset
       \else
         \noexpand\@elt{##1}%
       \fi}%
     \expandafter\xdef\csname cl@#2\endcsname
      {\csname cl@#2\endcsname}%
    \endgroup}}
```

(*End definition for* `\@removefromreset`*. This function is documented on page* **??**.)

`\@ifbothcounters` Test if arg #1 and #2 are counters and if so execute #3.

```
\def\@ifbothcounters#1#2#3{%
  \@ifundefined{c@#1}{\@nocounterr{#1}}%
    {% else counter is defined
     \@ifundefined{c@#2}{\@nocounterr{#2}}%
       {% else both counter and within  are defined
        #3}}}
```

(*End definition for* `\@ifbothcounters`*. This function is documented on page* **??**.)

`\counterwithout`

```
\def\counterwithout {\@ifstar\counterwithout@s\counterwithout@x}
```

```
\def\counterwithout@s#1#2{%
  \@ifbothcounters{#1}{#2}{\@removefromreset{#1}{#2}}}
```

```
\def\counterwithout@x#1#2{%
  \@ifbothcounters{#1}{#2}%
      {\@removefromreset{#1}{#2}%
       \expandafter
       \gdef\csname the#1\expandafter\endcsname\expandafter
            {\expandafter
             \@arabic\csname c@#1\endcsname}}}
```

(*End definition for* `\counterwithout`*. This function is documented on page* **??**.)

`\counterwithin`

```
\def\counterwithin{\@ifstar\counterwithin@s\counterwithin@x}
```

```
\def\counterwithin@s#1#2{%
  \@ifbothcounters{#1}{#2}{\@addtoreset{#1}{#2}}}
```

```
\def\counterwithin@x#1#2{%
  \@ifbothcounters{#1}{#2}%
      {\@addtoreset{#1}{#2}%
       \expandafter
       \gdef\csname the#1\expandafter\endcsname\expandafter
            {\csname the#2\expandafter\endcsname\expandafter
             .\expandafter
             \@arabic\csname c@#1\endcsname}}}
```

(*End definition for* `\counterwithin`*. This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}
⟨latexrelease⟩                  {\@removefromreset}{Add interfaces}%
⟨latexrelease⟩\let \@removefromreset \undefined
⟨latexrelease⟩\let \@ifbothcounters  \undefined
⟨latexrelease⟩\let \counterwithout   \undefined
⟨latexrelease⟩\let \counterwithout@s \undefined
⟨latexrelease⟩\let \counterwithout@x \undefined
⟨latexrelease⟩\let \counterwithin    \undefined
⟨latexrelease⟩\let \counterwithin@s  \undefined
⟨latexrelease⟩\let \counterwithin@x  \undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

Numbering commands for definitions of `\theCOUNTER` and `\list` arguments. All commands can now be used in text and math mode.

`\arabic` Representation of⟨*counter*⟩ as arabic numerals. Changed 29 Apr 86 to make it print the obvious thing it COUNTER not positive.

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

(*End definition for* `\arabic`*. This function is documented on page* **??**.)

`\roman` Representation of ⟨*counter*⟩ as lower-case Roman numerals.

```
\def\roman#1{\expandafter\@roman\csname c@#1\endcsname}
```

(*End definition for* `\roman`*. This function is documented on page* **??**.)

`\Roman` Representation of ⟨*counter*⟩ as upper-case Roman numerals.

```
\def\Roman#1{\expandafter\@Roman\csname c@#1\endcsname}
```

(*End definition for* `\Roman`*. This function is documented on page* **??**.)

`\alph` Representation of ⟨*counter*⟩ as a lower-case letter: 1 = a, 2 = b, etc.

```
\def\alph#1{\expandafter\@alph\csname c@#1\endcsname}
```

(*End definition for* `\alph`*. This function is documented on page* **??**.)

`\Alph` Representation of ⟨*counter*⟩ as an upper-case letter: 1 = A, 2 = B, etc.

```
\def\Alph#1{\expandafter\@Alph\csname c@#1\endcsname}
```

(*End definition for* `\Alph`*. This function is documented on page* **??**.)

`\fnsymbol` Representation of ⟨*COUNTER*⟩ as a footnote symbol: 1 = ∗, 2 = †, etc.

```
\def\fnsymbol#1{\expandafter\@fnsymbol\csname c@#1\endcsname}
```

(*End definition for* `\fnsymbol`*. This function is documented on page* **??**.)

`\@arabic` `\@arabic\FOOcounter` Representation of `\FOOcounter` as arabic numerals.

```
\def\@arabic#1{\number #1}  %% changed 29 Apr 86
```

(*End definition for* `\@arabic`*. This function is documented on page* **??**.)

\@roman \@roman\FOOcounter Representation of \FOOcounter as lower-case Roman numerals.

```
\def\@roman#1{\romannumeral #1}
```

(*End definition for* \@roman*. This function is documented on page* ??.)

\@Roman \@Roman\FOOcounter Representation of \FOOcounter as upper-case Roman numerals.

```
\def\@Roman#1{\expandafter\@slowromancap\romannumeral #1@}
```

(*End definition for* \@Roman*. This function is documented on page* ??.)

\@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
}
```

(*End definition for* \@slowromancap*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@alph*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@Alph*. This function is documented on page* ??.)

\@fnsymbol Typesetting old fashioned footnote symbols. This can be done both in text or math mode now.

This macro is another example of an ever recurring problem in TEX: Determining if something is text-mode or math-mode. It is imperative for the decision between text and math to be delayed until the actual typesetting is done as the code in question may go through an \edef or \write where an \ifmmode test would be executed prematurely. Hence in the implementation below, \@fnsymbol is not robust in itself but the parts doing the actual typesetting are.

In the case of \@fnsymbol we make use of the robust command \TextOrMath which takes two arguments and typesets the first if in text-mode and the second if in math-mode. Note that in order for this command to make the correct decision, it must insert a \relax token if run under regular TEX, which ruins any kerning between the preceding characters and whatever awaits typesetting. If you use eTEX as engine for LATEX (as recommended) this unfortunate side effect is not present.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@fnsymbol}{Use \TexOrMath}%
⟨*2ekernel | latexrelease⟩
\def\@fnsymbol#1{%
   \ifcase#1\or \TextOrMath\textasteriskcentered *\or
   \TextOrMath \textdagger \dagger\or
   \TextOrMath \textdaggerdbl \ddagger \or
   \TextOrMath \textsection  \mathsection\or
   \TextOrMath \textparagraph \mathparagraph\or
   \TextOrMath \textbardbl \|\or
   \TextOrMath {\textasteriskcentered\textasteriskcentered}{**}\or
   \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
   \TextOrMath {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
   \@ctrerr \fi
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@fnsymbol}{Use \TexOrMath}%
⟨latexrelease⟩\def\@fnsymbol#1{\ensuremath{%
⟨latexrelease⟩   \ifcase#1\or *\or \dagger\or \ddagger\or \mathsection\or
⟨latexrelease⟩     \mathparagraph\or \|\or **\or \dagger\dagger
⟨latexrelease⟩     \or \ddagger\ddagger \else\@ctrerr\fi}}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@fnsymbol`*. This function is documented on page* ??.)

`\TextOrMath` When using regular TEX, we make this command robust so that it always selects the correct branch in an `\ifmmode` switch with the usual disadvantage of ruining kerning. For the application we use it for here that shouldn't matter. The alternative would be to mimic `\IeC` from inputenc but then it wil have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of eTEX will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running eTEX but making sure not to permanently turn `\protected` into `\relax`.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\TextOrMath}{\TextOrMath}%
⟨*2ekernel | latexrelease⟩
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname protected\endcsname\relax
```

In case of ordinary TEX we define `\TextOrMath` as a robust command but make sure it always grabs its arguments. If we didn't do this it might very well gobble spaces in the input stream.

```
\DeclareRobustCommand\TextOrMath{%
  \ifmmode  \expandafter\@secondoftwo
  \else     \expandafter\@firstoftwo  \fi}
\protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
\else
```

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

```
\protected\expandafter\def\csname TextOrMath\space\endcsname{%
  \ifmmode  \expandafter\@secondoftwo
```

```
  \else     \expandafter\@firstoftwo  \fi}
\edef\TextOrMath#1#2{%
  \expandafter\noexpand\csname TextOrMath\space\endcsname
  {#1}{#2}}
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\TextOrMath}{\TextOrMath}%
⟨latexrelease⟩\let\TextOrMath\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\TextOrMath`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
```

File q
ltlength.dtx

1 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}}
```

(*End definition for* \newlength*. This function is documented on page* ??.)

\setlength

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\setlength}{Using \setlength with \dimen0}%
⟨*2ekernel | latexrelease⟩
\def\setlength#1#2{#1 #2\relax}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\setlength}{Using \setlength with \dimen0}%
⟨latexrelease⟩\def\setlength#1#2{#1#2\relax}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \setlength*. This function is documented on page* ??.)

\addtolength \relax added 24 Mar 86

```
\def\addtolength#1#2{\advance#1 #2\relax}
```

(*End definition for* \addtolength*. This function is documented on page* ??.)

\settoheight The obvious analogs of \settowidth.
\settodepth
\settowidth
\@settodim

```
\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}
\DeclareRobustCommand\settoheight{\@settodim\ht}
\DeclareRobustCommand\settodepth {\@settodim\dp}
\DeclareRobustCommand\settowidth {\@settodim\wd}
```

(*End definition for* \settoheight *and others. These functions are documented on page* ??.)

`\@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⟩

(*End definition for* `\@settopoint`*. This function is documented on page* **??**.)

File r
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.

1 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⟩
\def\@nomath#1{\relax\ifmmode
   \@font@warning{Command \noexpand#1invalid in math mode}\fi}
```

(*End definition for* `\@nomath`*. This function is documented on page* **??**.)

`\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.

```
\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}
```

(*End definition for* `\no@alphabet@error`*. This function is documented on page* **??**.)

`\new@mathgroup` `\mathgroup` We also give a new name to `\newfam` and `\fam` to avoid verbal confusion (see the introduction).[9]

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

(*End definition for* `\new@mathgroup` *and* `\mathgroup`*. These functions are documented on page* **??**.)

[9] 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.

2 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)

```
  \expandafter\endgroup
  \DeclareFontShape@}
```

(*End definition for* \DeclareFontShape. *This function is documented on page* ??.)

\DeclareFontShape@

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                    {\DeclareFontShape@}{Maybe drop one m}%
\def\DeclareFontShape@#1#2#3#4#5#6{%
  \expandafter\ifx\csname #1+#2\endcsname\relax
    \@latex@error{Font family ‘#1+#2’ unknown}\@eha
  \else
```

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

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

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

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

```
    \def\reserved@a{#6}%
    \global
    \expandafter\let\csname#5\expandafter\endcsname
       \ifx\reserved@a\@empty
         \@empty
       \else
         \reserved@a
       \fi
  \fi
 }
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\DeclareFontShape@}{Maybe drop one m}%
⟨latexrelease⟩
⟨latexrelease⟩\def\DeclareFontShape@#1#2#3#4#5#6{%
⟨latexrelease⟩   \expandafter\ifx\csname #1+#2\endcsname\relax
⟨latexrelease⟩     \@latex@error{Font family ‘#1+#2’ unknown}\@eha
⟨latexrelease⟩   \else
⟨latexrelease⟩     \expandafter
⟨latexrelease⟩       \xdef\csname#1/#2/#3/#4\endcsname{\expandafter\noexpand
⟨latexrelease⟩                                 \csname #5\endcsname}%
⟨latexrelease⟩     \def\reserved@a{#6}%
⟨latexrelease⟩     \global
⟨latexrelease⟩     \expandafter\let\csname#5\expandafter\endcsname
⟨latexrelease⟩        \ifx\reserved@a\@empty
⟨latexrelease⟩          \@empty
⟨latexrelease⟩        \else
⟨latexrelease⟩          \reserved@a
⟨latexrelease⟩        \fi
⟨latexrelease⟩   \fi
⟨latexrelease⟩  }
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\DeclareFontShape@`*. This function is documented on page* **??**.)

`\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
  }
```

(*End definition for* `\DeclareFixedFont`*. This function is documented on page* **??**.)

`\do@subst@correction`

```
\def\do@subst@correction{%
       \xdef\subst@correction{%
          \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
}
```

(*End definition for* `\do@subst@correction`*. This function is documented on page* **??**.)

\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
    }%
}
```

(*End definition for* \DeclareFontFamily. *This function is documented on page* ??.)

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

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

(*End definition for* \cdp@list. *This function is documented on page* ??.)

\cdp@elt

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

(*End definition for* \cdp@elt. *This function is documented on page* ??.)

\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@
```

(*End definition for* `\DeclareFontEncoding`*. This function is documented on page* ??.)

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

```
\def\LastDeclaredEncoding{}
```

(*End definition for* `\LastDeclaredEncoding`*. This function is documented on page* ??.)

`\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
```

(*End definition for* \DeclareFontSubstitution*. This function is documented on page* **??**.)

\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
```

(*End definition for* \DeclareFontEncodingDefaults*. This function is documented on page* **??**.)

\default@T
\default@M

```
\let\default@T\@empty
\let\default@M\@empty
```

(*End definition for* \default@T *and* \default@M*. These functions are documented on page* **??**.)

\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`[10] 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.

```
          \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
```

(*End definition for* `\DeclarePreloadSizes`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\ifmath@fonts`*. This function is documented on page* **??**.)

[10] We cannot use `\@tempa` since it is needed in `\pickup@font`.

\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
```

(*End definition for* \DeclareMathSizes *and* \DeclareMathSizes*. *These functions are documented on page* ??.)

\@DeclareMathSizes

This modification by Michael J. Downes on comp.text.tex on 2002/10/17 allows the user to have settings such as
\DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@DeclareMathSizes}%
⟨latexrelease⟩                 {Arbitrary units in \DeclareMathSizes}%
⟨*2ekernel | latexrelease⟩
\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
    \@defaultunits\dimen@ii #3pt\relax\@nnil
    \@defaultunits\@tempdima #4pt\relax\@nnil
    \@defaultunits\@tempdimb #5pt\relax\@nnil
    \toks@{#1}%
    \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
      \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
      \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
      \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
      \the\toks@
    }%
  \fi
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@DeclareMathSizes}%
⟨latexrelease⟩                 {Arbitrary units in \DeclareMathSizes}%
⟨latexrelease⟩\def\@DeclareMathSizes#1#2#3#4#5{%
⟨latexrelease⟩    \@defaultunits\dimen@#2pt\relax\@nnil
⟨latexrelease⟩    \if$#3$%
⟨latexrelease⟩      \expandafter \let
⟨latexrelease⟩        \csname S@\strip@pt\dimen@\endcsname
⟨latexrelease⟩        \math@fontsfalse
⟨latexrelease⟩    \else
⟨latexrelease⟩      \expandafter \gdef
⟨latexrelease⟩      \csname S@\strip@pt\dimen@\endcsname
⟨latexrelease⟩            {\gdef\tf@size{#3}\gdef\sf@size{#4}%
⟨latexrelease⟩                              \gdef\ssf@size{#5}%
⟨latexrelease⟩             #1%
⟨latexrelease⟩                              }%
⟨latexrelease⟩    \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\@DeclareMathSizes
```

(*End definition for* `\@DeclareMathSizes`*. This function is documented on page* **??**.)

3 Selecting a new font

3.1 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` in-between 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
}
```

(*End definition for* `\fontencoding` *and* `\f@encoding`*. These functions are documented on page* **??**.)

`\@@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
}
```

(*End definition for* \@@enc@update. *This function is documented on page* ??.)

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

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

(*End definition for* \enc@update. *This function is documented on page* ??.)

\fontfamily
\f@family
\fontseries
\f@series
\fontshape
\f@shape

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

There are now defined later (and differently).

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

(*End definition for* \fontfamily *and others. These functions are documented on page* ??.)

\usefont Some handy abbreviation if you want to get some particular font in the current size. If also the size should change one has to issue a \fontsize command first.

\fontencoding needs to do some setup work so we call that, but instead of calling \fontfamily, \fontseries and \fontshape it earlier versions of this code did, we now set \f@family, etc. directly. If we would call \fontseries or \fontshape as it was done in the past, they would now interact with the existing series and shape which is not desired if we intend to use an explicit font shape!

```
\DeclareRobustCommand\usefont[4]{\fontencoding{#1}%
   \edef\f@family{#2}%
   \edef\f@series{#3}%
   \edef\f@shape{#4}\selectfont
   \ignorespaces}
```

(*End definition for* \usefont. *This function is documented on page* ??.)

\linespread The command \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}
```

(*End definition for* \linespread. *This function is documented on page* ??.)

\fontsize We also define a macro that allows to specify a size. In this case, however, we also need the value of \baselineskip. As the first argument to \set@fontsize we pass the current value of \baselinestretch. This will either match the internal value (in which case nothing changes, or it will be an updated value due to a user change of that macro using \renewcommand. If we would pass the internal \f@linespread such a change would be effectively overwritten by a size change.

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

(*End definition for* \fontsize. *This function is documented on page* ??.)

\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
```

(*End definition for* \f@linespread. *This function is documented on page* ??.)

\cf@encoding

```
\let\f@encoding\@empty
\let\cf@encoding\@empty
```

(*End definition for* \cf@encoding. *This function is documented on page* ??.)

\@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}
```

(*End definition for* \@defaultunits. *This function is documented on page* ??.)

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

```
\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}
```

(*End definition for* \strip@pt *and* \rem@pt. *These functions are documented on page* ??.)

\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}
```

(*End definition for* \mathversion *and* \math@version*. These functions are documented on page* ??.)

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_ε 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
```

(*End definition for* \frozen@everymath *and* \frozen@everydisplay*. These functions are documented on page* ??.)

\everymath
\everydisplay

Now we provide now user hooks that will be called in the frozen internals.

```
\newtoks\everymath
\newtoks\everydisplay
```

(*End definition for* \everymath *and* \everydisplay*. These functions are documented on page* ??.)

\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}
```

(*End definition for* \frozen@everymath*. This function is documented on page* ??.)

\frozen@everydisplay

Ditto for the display hook.

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

(*End definition for* \frozen@everydisplay*. This function is documented on page* ??.)

\curr@math@size

This holds locally the current math size.

```
\let\curr@math@size\@empty
```

(*End definition for* \curr@math@size*. This function is documented on page* ??.)

3.2 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}
```

(*End definition for* \pickup@font*. This function is documented on page* ??.)

\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}}}
```

(*End definition for* \split@name*. This function is documented on page* ??.)

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

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

(*End definition for* \curr@fontshape*. This function is documented on page* ??.)

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

```
\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}
```

```
\def\try@load@fontshape{%
   \expandafter
   \ifx\csname \f@encoding+\f@family\endcsname\relax
     \@font@info{Trying to load 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}
```

(*End definition for* \define@newfont*. This function is documented on page* ??.)

\nfss@catcodes This macro should contain the standard \catcode assignments to all characters which are used in the commands found in an .fd file and which might have special \catcodes in the middle of a document. If necessary, this list can be extended in a package file using a suitable number of \expandafter, i.e.,

```
\expandafter\def\expandafter\nfss@catcodes
      \expandafter{\nfss@catcodes <additional settings>}
```

Note, that this macro might get executed several times since it is also called by `\DeclareFontShape`, thus it probably should not be misused as a general purpose hook.

```
\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\"%
}
```

(*End definition for* `\nfss@catcodes`*. This function is documented on page* **??**.)

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

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\LoadFontDefinitionFile}{Loading .fd files}%
\def\LoadFontDefinitionFile#1#2{%
  \begingroup
    \edef\f@encoding{#1}%
    \edef\f@family{#2}%
    \try@load@fontshape
  \endgroup
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\LoadFontDefinitionFile}{Loading .fd files}%
⟨latexrelease⟩
```

```
⟨latexrelease⟩\let\LoadFontDefinitionFile\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\LoadFontDefinitionFile`*. This function is documented on page* **??**.)

`\DeclareFontFamilySubstitution`

The idea for this macro is stolen from the `substitutefont` package by Günter Milde, with some modifications and a new name.

Its purpose is to provide characters in a special encoding tht are not available in the current font family to be taken from a different family that is visually compatible (or not if you choose badly). For example, you can match the GFS Didot Greek characters with TEX Gyre Pagella (Palatino) by specifying

```
\DeclareFontFamilySubstitution{LGR}{qpl}{udidot}
```

This way if you ask for the `LGR` encoding in for the `qpl` family you get the characters from the `udidot` family substituted.

We need to ensure that the macro is defined with `\nfss@catcodes` in force (not quite sure why at the moment to be honest).

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩        {\DeclareFontFamilySubstitution}{Provide family substituation}%
\begingroup
\nfss@catcodes
\gdef\DeclareFontFamilySubstitution#1#2#3{%
```

We only provide a set of silent substitutions. The package also (re)declared the family, but this is incorrect in my eyes and it is better to handle that differently.

Of course the families may still need loading at this point and so we arange for this. Otherwise we might run into trouble because the necessary `\DeclareFontFamily` has not been seen.

```
    \LoadFontDefinitionFile{#1}{#2}%
    \LoadFontDefinitionFile{#1}{#3}%
```

```
    \DeclareFontShape{#1}{#2}{m}{it}{<->ssub * #3/m/it}{}%
    \DeclareFontShape{#1}{#2}{m}{n}{<->ssub * #3/m/n}{}%
    \DeclareFontShape{#1}{#2}{m}{sc}{<->ssub * #3/m/sc}{}%
    \DeclareFontShape{#1}{#2}{m}{sl}{<->ssub * #3/m/sl}{}%
```

These days a few more shapes might be around, so we declare those too. If they don't exist then after the first substitution normal fallbacks will happen.

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

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

```
    \DeclareFontShape{#1}{#2}{b}{it}{<->ssub * #3/b/it}{}%
    \DeclareFontShape{#1}{#2}{b}{n}{<->ssub * #3/b/n}{}%
    \DeclareFontShape{#1}{#2}{b}{scit}{<->ssub * #3/b/scit}{}%
    \DeclareFontShape{#1}{#2}{b}{scsl}{<->ssub * #3/b/scsl}{}%
    \DeclareFontShape{#1}{#2}{b}{sc}{<->ssub * #3/b/sc}{}%
    \DeclareFontShape{#1}{#2}{b}{sl}{<->ssub * #3/b/sl}{}%
    \DeclareFontShape{#1}{#2}{b}{sw}{<->ssub * #3/b/sw}{}%
    \DeclareFontShape{#1}{#2}{bx}{it}{<->ssub * #3/bx/it}{}%
```

```
   \DeclareFontShape{#1}{#2}{bx}{n}{<->ssub * #3/bx/n}{}%
   \DeclareFontShape{#1}{#2}{bx}{scit}{<->ssub * #3/bx/scit}{}%
   \DeclareFontShape{#1}{#2}{bx}{scsl}{<->ssub * #3/bx/scsl}{}%
   \DeclareFontShape{#1}{#2}{bx}{sc}{<->ssub * #3/bx/sc}{}%
   \DeclareFontShape{#1}{#2}{bx}{sl}{<->ssub * #3/bx/sl}{}%
   \DeclareFontShape{#1}{#2}{bx}{sw}{<->ssub * #3/bx/sw}{}%
}
\endgroup
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩      {\DeclareFontFamilySubstitution}{Provide family substituation}%
⟨latexrelease⟩
⟨latexrelease⟩\let\DeclareFontFamilySubstitution\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\DeclareFontFamilySubstitution`*. This function is documented on page* ??.)

`\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!

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                 {\DeclareErrorFont}{No side effects please}%
\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—and now it did.

```
%       \gdef\f@encoding{#1}%
      \gdef\default@family{#2}%
      \gdef\default@series{#3}%
      \gdef\default@shape{#4}%
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\DeclareErrorFont}{No side effects please}%
⟨latexrelease⟩
⟨latexrelease⟩\def\DeclareErrorFont#1#2#3#4#5{%
⟨latexrelease⟩      \xdef\error@fontshape{%
⟨latexrelease⟩          \noexpand\expandafter\noexpand\split@name\noexpand\string
⟨latexrelease⟩          \expandafter\noexpand\csname#1/#2/#3/#4/#5\endcsname
⟨latexrelease⟩          \noexpand\@nil}%
⟨latexrelease⟩      \gdef\default@family{#2}%
```

```
⟨latexrelease⟩        \gdef\default@series{#3}%
⟨latexrelease⟩        \gdef\default@shape{#4}%
⟨latexrelease⟩        \global\let\f@family\default@family
⟨latexrelease⟩        \global\let\f@series\default@series
⟨latexrelease⟩        \global\let\f@shape\default@shape
⟨latexrelease⟩        \gdef\f@size{#5}%
⟨latexrelease⟩        \gdef\f@baselineskip{#5pt}%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@onlypreamble\DeclareErrorFont
```

(*End definition for* `\DeclareErrorFont`*. This function is documented on page* **??**.)

`\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!

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\wrong@fontshape}%
⟨latexrelease⟩                  {Font substituation in preamble}%
⟨*2ekernel | latexrelease⟩
\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
```

If we change the font family and we are in the preamble then the corresponding `.fd` file may not been loaded yet. Therefore we try this now. Otherwise equating the requested font shape with the finally selected fontshape below will fail and can result in "NFSS tables corruped". After begin document that will not happen as all `.fd` files involved in substituation are loaded at `\begin{document}`.

```
            \begingroup
               \try@load@fontshape
            \endgroup
         \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}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\wrong@fontshape}%
⟨latexrelease⟩                  {Font substituation in preamble}%
⟨latexrelease⟩\def\wrong@fontshape{%
⟨latexrelease⟩    \csname D@\f@encoding\endcsname
⟨latexrelease⟩    \edef\reserved@a{\csname\curr@fontshape\endcsname}%
⟨latexrelease⟩  \ifx\last@fontshape\reserved@a
⟨latexrelease⟩     \errmessage{Corrupted NFSS tables}%
⟨latexrelease⟩     \error@fontshape
⟨latexrelease⟩  \else
⟨latexrelease⟩    \let\f@shape\default@shape
⟨latexrelease⟩    \expandafter\ifx\csname\curr@fontshape\endcsname\relax
⟨latexrelease⟩       \let\f@series\default@series
⟨latexrelease⟩        \expandafter
```

```
⟨latexrelease⟩          \ifx\csname\curr@fontshape\endcsname\relax
⟨latexrelease⟩           \let\f@family\default@family
⟨latexrelease⟩        \fi \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩     \@font@warning{Font shape
⟨latexrelease⟩            `\expandafter\string\reserved@a'
⟨latexrelease⟩            \expandafter\@gobble\string\@undefined
⟨latexrelease⟩            \MessageBreak
⟨latexrelease⟩            using `\curr@fontshape' instead\@wrong@font@char}%
⟨latexrelease⟩    \global\let\last@fontshape\reserved@a
⟨latexrelease⟩    \gdef\@defaultsubs{%
⟨latexrelease⟩      \@font@warning{Some font shapes were not available,
⟨latexrelease⟩                      defaults substituted.\@gobbletwo}}%
⟨latexrelease⟩    \global\expandafter\expandafter\expandafter\let
⟨latexrelease⟩       \expandafter\reserved@a
⟨latexrelease⟩           \csname\curr@fontshape\endcsname
⟨latexrelease⟩    \xdef\font@name{%
⟨latexrelease⟩      \csname\curr@fontshape/\f@size\endcsname}%
⟨latexrelease⟩    \pickup@font}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\wrong@fontshape`. *This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@wrong@font@char`. *This function is documented on page* **??**.)

`\@@defaultsubs` `\@defaultsubs` See above.

```
\let\@defaultsubs\relax
```

(*End definition for* `\@@defaultsubs` *and* `\@defaultsubs`. *These functions are documented on page* **??**.)

`\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>{}
```

(*End definition for* `\strip@prefix`. *This function is documented on page* **??**.)

4 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
```

(*End definition for* `\install@mathalphabet`. *This function is documented on page* **??**.)

`\math@fonts`

```
\let\math@fonts\@empty
```

(*End definition for* `\math@fonts`. *This function is documented on page* **??**.)

\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 of `\math@egroup` if the `margid` option is in force) which would swallow our closing `\fi`. So we use the `\expandafter` technique to remove the `\fi` before the `\use@mathgroup` is expanded.

```
%\expandafter #1\fi}
```

(*End definition for* `\select@group`*. This function is documented on page* **??**.)

`\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
      }
```

(*End definition for* `\extract@alph@from@version`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\math@bgroup` *and* `\math@egroup`*. These functions are documented on page* **??**.)

`\calculate@math@sizes`

Here is the default definition for `\calculate@math@sizes` a more elaborate interface is under testing in mthscale.sty.

```
\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}}
```

(*End definition for* `\calculate@math@sizes`*. This function is documented on page* **??**.)

`\defaultscriptratio` `\defaultscriptscriptratio`

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

```
\def\defaultscriptratio{.7}
\def\defaultscriptscriptratio{.5}
```

(*End definition for* `\defaultscriptratio` *and* `\defaultscriptscriptratio`*. These functions are documented on page* **??**.)

`\noaccents@`

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

```
\ifx\noaccents@\@undefined
  \let\noaccents@\@empty
\fi
```

(*End definition for* `\noaccents@`*. This function is documented on page* **??**.)

\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⟩
⟨latexrelease⟩\IncludeInRelease{2017/01/01}{\showhyphens}%
⟨latexrelease⟩                  {XeTeX support for \showhyphens}%
⟨*2ekernel | latexrelease⟩
\ifx\XeTeXcharclass\@undefined
```

Version for engines other than XeTeX.

```
\DeclareRobustCommand\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}}
\else
```

XeTeX version. When using system fonts XeTeX reports consecutive runs of characters as a single item in box logging, which means the standard \showhyphens does not work. This version typesets the text into a narrow box to force hyphenation and then reconstructs a horizontal list with explicit hyphens to generate the display. Note that the lmr OpenType font is forced, this works even if the characters are not in the font as hyphenation is attempted due to the width of the space and hyphen character. It may generate spurious Missing Character warnings in the log, these are however suppressed from the terminal output by ensuring that \tracingonline is locally zero.

```
\DeclareRobustCommand\showhyphens[1]{%
  \setbox0\vbox{%
    \usefont{TU}{lmr}{m}{n}%
    \hsize 1sp %
    \hbadness\@M
    \hfuzz\maxdimen
    \tracingonline\z@
    \everypar={}%
    \leftskip\z@skip
    \rightskip\z@skip
    \parfillskip\z@skip
    \hyphenpenalty=-\@M
    \pretolerance\m@ne
    \interlinepenalty\z@
    \clubpenalty\z@
    \widowpenalty\z@
    \brokenpenalty1127 %
    \setbox\z@\hbox{}%
    \noindent
    \hskip\z@skip
    #1%
    \par
```

Note here we stop the loop if made no progress, non-removable items may mean that we can not process the whole list (which would be testable as \lastnodetype=-1).

```
     \loop
     \@tempswafalse
```

```
      \ifnum\lastnodetype=11\unskip\@tempswatrue\fi
      \ifnum\lastnodetype=12\unkern\@tempswatrue\fi
      \ifnum\lastnodetype=13 %
       \count@\lastpenalty
       \unpenalty\@tempswatrue
     \fi
     \ifnum\lastnodetype=\@ne
      \setbox\tw@\lastbox\@tempswatrue
      \setbox0\hbox{\unhbox\tw@\unskip\unskip\unpenalty
                    \ifnum\count@=1127 \else\ \fi
                    \unhbox0}%
      \count@\z@
     \fi
     \if@tempswa
     \repeat
    \hbadness\z@
    \hsize\maxdimen
    \showboxdepth\z@
    \tolerance\m@ne
    \hyphenpenalty\z@
    \noindent\unhbox\z@
}}
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\showhyphens}%
⟨latexrelease⟩                  {XeTeX support for \showhyphens}%
⟨latexrelease⟩\gdef\showhyphens#1{%
⟨latexrelease⟩   \setbox0\vbox{%
⟨latexrelease⟩     \color@begingroup
⟨latexrelease⟩     \everypar{}%
⟨latexrelease⟩     \parfillskip\z@skip\hsize\maxdimen
⟨latexrelease⟩     \normalfont
⟨latexrelease⟩     \pretolerance\m@ne\tolerance\m@ne
⟨latexrelease⟩     \hbadness\z@\showboxdepth\z@\ #1%
⟨latexrelease⟩     \color@endgroup}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\showhyphens`*. This function is documented on page* **??**.)

`\addto@hook` We need a macro to add tokens to a hook.

```
\long\def\addto@hook#1#2{#1\expandafter{\the#1#2}}
```

(*End definition for* `\addto@hook`*. This function is documented on page* **??**.)

`\@vpt`

```
\def\@vpt{5}
```

(*End definition for* `\@vpt`*. This function is documented on page* **??**.)

`\@vipt`

```
\def\@vipt{6}
```

(*End definition for* `\@vipt`*. This function is documented on page* **??**.)

`\@viipt`

```
\def\@viipt{7}
```

(*End definition for* `\@viipt`*. This function is documented on page* **??**.)

`\@viiipt`

```
\def\@viiipt{8}
```

(*End definition for* `\@viiipt`*. This function is documented on page* **??**.)

`\@ixpt`

```
\def\@ixpt{9}
```

(*End definition for* `\@ixpt`*. This function is documented on page* **??**.)

`\@xpt`

```
\def\@xpt{10}
```

(*End definition for* `\@xpt`*. This function is documented on page* **??**.)

`\@xipt`

```
\def\@xipt{10.95}
```

(*End definition for* `\@xipt`*. This function is documented on page* **??**.)

`\@xiipt`

```
\def\@xiipt{12}
```

(*End definition for* `\@xiipt`*. This function is documented on page* **??**.)

`\@xivpt`

```
\def\@xivpt{14.4}
```

(*End definition for* `\@xivpt`*. This function is documented on page* **??**.)

`\@xviipt`

```
\def\@xviipt{17.28}
```

(*End definition for* `\@xviipt`*. This function is documented on page* **??**.)

`\@xxpt`

```
\def\@xxpt{20.74}
```

(*End definition for* `\@xxpt`*. This function is documented on page* **??**.)

`\@xxvpt`

```
\def\@xxvpt{24.88}
```

(*End definition for* `\@xxvpt`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
```

File s
ltfssaxes.dtx

This file contains the implementation for handling extra axes splitting the series and the values into sub-categories. selection commands. See other parts of the LaTeX distribution, or *The LaTeX Companion* for higher level documentation of the LaTeX Font Selection Scheme.

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

```
⟨*2ekernel⟩
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩    {\DeclareFontSeriesChangeRule}{Series change rules}%
```

1 Changing the font series

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

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

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

1.1 The series lookup table

`\DeclareFontSeriesChangeRule`

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

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

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

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

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

(*End definition for* `\DeclareFontSeriesChangeRule`*. This function is documented on page* **??**.)

1.2 Mapping rules for series changes

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

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

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

Design considerations for mapping entries:

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

  In particular this is also true for cases involving `m`, e.g., `bm` (bold medium width) which automatially gets reduced result in `b` or `mc` (medium weight condensed) which becomes `c` as a result.
- Only a few entries have "alterative" values and perhaps most of them should get dropped. Or maybe not ... needs some thought perhaps.

  The idea is that you don't want the normal substitution to kick in because that would reset the shape first and it may be better to stay with `b` when a change to `c` is requested and `bc` doesn't exist, than to go to first change the shape to `n` and then find that `bc/n` doesn't exist either and thus ending up wth `m/n`.
- Also: while I did set up all nine standard weight values from `ul` to `ub` I only bothered to provide entries for `ec`, `sc`, `c` and `x`, because other levels of compression/expansion are not in any real fonts that I know.

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

```
\DeclareFontSeriesChangeRule {bc}{b}{bc}{}
\DeclareFontSeriesChangeRule {bc}{c}{bc}{}
\DeclareFontSeriesChangeRule {bc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {bc}{ec}{bec} {bc}
\DeclareFontSeriesChangeRule {bc}{el}{elc}{}
\DeclareFontSeriesChangeRule {bc}{l}{lc}{}
\DeclareFontSeriesChangeRule {bc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {bc}{sc}{bsc} {bc}
\DeclareFontSeriesChangeRule {bc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {bc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {bc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {bc}{x}{bx}{}
```

```
\DeclareFontSeriesChangeRule {bx}{b}{bx}{}
\DeclareFontSeriesChangeRule {bx}{c}  {bc}  {bx} %<-----
\DeclareFontSeriesChangeRule {bx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {bx}{ec}  {bec}  {bx} %<-----
\DeclareFontSeriesChangeRule {bx}{el}{elx}{}
\DeclareFontSeriesChangeRule {bx}{l}{lx}{}
\DeclareFontSeriesChangeRule {bx}{sb} {sbx} {}
\DeclareFontSeriesChangeRule {bx}{sc}  {bsc}  {bx} %<-----
\DeclareFontSeriesChangeRule {bx}{sl}{slx}  {}
\DeclareFontSeriesChangeRule {bx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {bx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {bx}{x}{bx}{}

\DeclareFontSeriesChangeRule {b}{bx} {bx}  {b} %<-----
\DeclareFontSeriesChangeRule {b}{c}  {bc}  {b} %<-----
\DeclareFontSeriesChangeRule {b}{ec}  {bec}  {b} %<-----
\DeclareFontSeriesChangeRule {b}{sb} {sb}  {b} %<-----
\DeclareFontSeriesChangeRule {b}{sc}  {bsc}  {b} %<-----
\DeclareFontSeriesChangeRule {b}{x}  {bx}  {b} %<-----

\DeclareFontSeriesChangeRule {c}{bx} {bx}  {b}  %<-----
\DeclareFontSeriesChangeRule {c}{b}{bc}{}
\DeclareFontSeriesChangeRule {c}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {c}{el}{elc}{}
\DeclareFontSeriesChangeRule {c}{l}{lc}{}
\DeclareFontSeriesChangeRule {c}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {c}{sl}{slc}{}
\DeclareFontSeriesChangeRule {c}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {c}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {c}{x}{x}{m}               %<-----

\DeclareFontSeriesChangeRule {ebc}{b}{bc}{}
\DeclareFontSeriesChangeRule {ebc}{c}{ebc}{}
\DeclareFontSeriesChangeRule {ebc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {ebc}{ec}{ebec}{ebc}
\DeclareFontSeriesChangeRule {ebc}{el}{elc}{}
\DeclareFontSeriesChangeRule {ebc}{l}{lc}{}
\DeclareFontSeriesChangeRule {ebc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {ebc}{sc}{ebsc}{ebc}
\DeclareFontSeriesChangeRule {ebc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {ebc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {ebc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {ebc}{x}{ebx}{}

\DeclareFontSeriesChangeRule {ec}{bx} {bx}  {b}  %<-----
\DeclareFontSeriesChangeRule {ec}{b}{bec}{}
\DeclareFontSeriesChangeRule {ec}{eb}{ebec}{}
\DeclareFontSeriesChangeRule {ec}{el}{elec}{}
\DeclareFontSeriesChangeRule {ec}{l}{lec}{}
\DeclareFontSeriesChangeRule {ec}{sb}{sbec}{}
\DeclareFontSeriesChangeRule {ec}{sl}{slec}{}
\DeclareFontSeriesChangeRule {ec}{ub}{ubec}{}
\DeclareFontSeriesChangeRule {ec}{ul}{ulec}{}
\DeclareFontSeriesChangeRule {ec}{x}{x}{m}              %<-----

\DeclareFontSeriesChangeRule {sc}{bx} {bx}  {b}  %<-----
\DeclareFontSeriesChangeRule {sc}{b}{bsc}{}
```

```
\DeclareFontSeriesChangeRule {sc}{eb}{ebsc}{}
\DeclareFontSeriesChangeRule {sc}{el}{elsc}{}
\DeclareFontSeriesChangeRule {sc}{l}{lsc}{}
\DeclareFontSeriesChangeRule {sc}{sb}{sbsc}{}
\DeclareFontSeriesChangeRule {sc}{sl}{slsc}{}
\DeclareFontSeriesChangeRule {sc}{ub}{ubsc}{}
\DeclareFontSeriesChangeRule {sc}{ul}{ulsc}{}
\DeclareFontSeriesChangeRule {sc}{x}{x}{m}                %<-----

\DeclareFontSeriesChangeRule {ebx}{b}{bx}{}
\DeclareFontSeriesChangeRule {ebx}{c}{ebc}{}
\DeclareFontSeriesChangeRule {ebx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {ebx}{ec}{ebec}{}
\DeclareFontSeriesChangeRule {ebx}{el}{elx}{}
\DeclareFontSeriesChangeRule {ebx}{l}{lx}{}
\DeclareFontSeriesChangeRule {ebx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {ebx}{sc}{ebsc}{}
\DeclareFontSeriesChangeRule {ebx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {ebx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {ebx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {ebx}{x}{ebx}{}

\DeclareFontSeriesChangeRule {eb}{c}{ebc}{}
\DeclareFontSeriesChangeRule {eb}{ec}{ebec}{}
\DeclareFontSeriesChangeRule {eb}{sc}{ebsc}{}
\DeclareFontSeriesChangeRule {eb}{x}{ebx}{}

\DeclareFontSeriesChangeRule {elc}{b}{bc}{}
\DeclareFontSeriesChangeRule {elc}{c}{elc}{}
\DeclareFontSeriesChangeRule {elc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {elc}{ec}{elec}{}
\DeclareFontSeriesChangeRule {elc}{el}{elc}{}
\DeclareFontSeriesChangeRule {elc}{l}{lc}{}
\DeclareFontSeriesChangeRule {elc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {elc}{sc}{elsc}{}
\DeclareFontSeriesChangeRule {elc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {elc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {elc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {elc}{x}{elx}{}

\DeclareFontSeriesChangeRule {elx}{b}{bx}{}
\DeclareFontSeriesChangeRule {elx}{c}{elc}{}
\DeclareFontSeriesChangeRule {elx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {elx}{ec}{elec}{}
\DeclareFontSeriesChangeRule {elx}{el}{elx}{}
\DeclareFontSeriesChangeRule {elx}{l}{lx}{}
\DeclareFontSeriesChangeRule {elx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {elx}{sc}{elsc}{}
\DeclareFontSeriesChangeRule {elx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {elx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {elx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {elx}{x}{elx}{}

\DeclareFontSeriesChangeRule {el}{c}{elc}{}
\DeclareFontSeriesChangeRule {el}{ec}{elec}{}
\DeclareFontSeriesChangeRule {el}{sc}{elsc}{}
\DeclareFontSeriesChangeRule {el}{x}{elx}{}
```

```
\DeclareFontSeriesChangeRule {lc}{b}{bc}{}
\DeclareFontSeriesChangeRule {lc}{c}{lc}{}
\DeclareFontSeriesChangeRule {lc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {lc}{ec}{lec}{}
\DeclareFontSeriesChangeRule {lc}{el}{elc}{}
\DeclareFontSeriesChangeRule {lc}{l}{lc}{}
\DeclareFontSeriesChangeRule {lc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {lc}{sc}{lsc}{}
\DeclareFontSeriesChangeRule {lc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {lc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {lc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {lc}{x}{lx}{}

\DeclareFontSeriesChangeRule {lx}{b}{bx}{}
\DeclareFontSeriesChangeRule {lx}{c}{lc}{}
\DeclareFontSeriesChangeRule {lx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {lx}{ec}{lec}{}
\DeclareFontSeriesChangeRule {lx}{el}{elx}{}
\DeclareFontSeriesChangeRule {lx}{l}{lx}{}
\DeclareFontSeriesChangeRule {lx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {lx}{sc}{lsc}{}
\DeclareFontSeriesChangeRule {lx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {lx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {lx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {lx}{x}{lx}{}

\DeclareFontSeriesChangeRule {l}{bx} {bx}  {b} %<-----
\DeclareFontSeriesChangeRule {l}{b}  {b}   {bx} %<-----
\DeclareFontSeriesChangeRule {l}{c}  {lc}  {l}  % ? %<-----
\DeclareFontSeriesChangeRule {l}{ec}  {lec}  {l}  % ? %<-----
\DeclareFontSeriesChangeRule {l}{sb} {sb}  {b}  % ? %<-----
\DeclareFontSeriesChangeRule {l}{sc}  {lsc}  {l}  % ? %<-----
\DeclareFontSeriesChangeRule {l}{x}  {lx}  {l}  % ? %<-----

\DeclareFontSeriesChangeRule {m}{bx} {bx} {b}    %<-----
\DeclareFontSeriesChangeRule {m}{b}  {b}  {bx} %<-----
\DeclareFontSeriesChangeRule {m}{c}  {c}  {m} %<-----
\DeclareFontSeriesChangeRule {m}{ec}  {ec}  {m} %<-----
\DeclareFontSeriesChangeRule {m}{l}  {l}  {m} %<-----
\DeclareFontSeriesChangeRule {m}{sb} {sb} {b} %<-----
\DeclareFontSeriesChangeRule {m}{sc}  {sc}  {m} %<-----
\DeclareFontSeriesChangeRule {m}{x}  {x}  {m} %<-----

\DeclareFontSeriesChangeRule {sbc}{b}{bc}{}
\DeclareFontSeriesChangeRule {sbc}{c}{sbc}{}
\DeclareFontSeriesChangeRule {sbc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {sbc}{ec}{sbec}{sbc}
\DeclareFontSeriesChangeRule {sbc}{el}{elc}{}
\DeclareFontSeriesChangeRule {sbc}{l}{lc}{}
\DeclareFontSeriesChangeRule {sbc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {sbc}{sc}{sbsc}{sbc}
\DeclareFontSeriesChangeRule {sbc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {sbc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {sbc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {sbc}{x}{sbx}{}

\DeclareFontSeriesChangeRule {sbx}{b}{bx}{}
```

```
\DeclareFontSeriesChangeRule {sbx}{c}{sbc}{}
\DeclareFontSeriesChangeRule {sbx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {sbx}{ec}{sbec}{}
\DeclareFontSeriesChangeRule {sbx}{el}{elx}{}
\DeclareFontSeriesChangeRule {sbx}{l}{lx}{}
\DeclareFontSeriesChangeRule {sbx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {sbx}{sc}{sbsc}{}
\DeclareFontSeriesChangeRule {sbx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {sbx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {sbx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {sbx}{x}{sbx}{}

\DeclareFontSeriesChangeRule {sb}{c}  {sbc} {bc} %? %<-----
\DeclareFontSeriesChangeRule {sb}{ec}  {sbec} {sbc} %? %<-----
\DeclareFontSeriesChangeRule {sb}{sc}  {sbsc} {sbc} %? %<-----
\DeclareFontSeriesChangeRule {sb}{x}  {sbx} {bx} %? %<-----

\DeclareFontSeriesChangeRule {slc}{b}{bc}{}
\DeclareFontSeriesChangeRule {slc}{c}{slc}{}
\DeclareFontSeriesChangeRule {slc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {slc}{ec}{slec}{}
\DeclareFontSeriesChangeRule {slc}{el}{elc}{}
\DeclareFontSeriesChangeRule {slc}{l}{lc}{}
\DeclareFontSeriesChangeRule {slc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {slc}{sc}{slsc}{}
\DeclareFontSeriesChangeRule {slc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {slc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {slc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {slc}{x}{slx}{}

\DeclareFontSeriesChangeRule {slx}{b}{bx}{}
\DeclareFontSeriesChangeRule {slx}{c}{slc}{}
\DeclareFontSeriesChangeRule {slx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {slx}{ec}{slec}{}
\DeclareFontSeriesChangeRule {slx}{el}{elx}{}
\DeclareFontSeriesChangeRule {slx}{l}{lx}{}
\DeclareFontSeriesChangeRule {slx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {slx}{sc}{slsc}{}
\DeclareFontSeriesChangeRule {slx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {slx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {slx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {slx}{x}{slx}{}

\DeclareFontSeriesChangeRule {sl}{c}{slc}{}
\DeclareFontSeriesChangeRule {sl}{ec}{slec}{}
\DeclareFontSeriesChangeRule {sl}{sc}{slsc}{}
\DeclareFontSeriesChangeRule {sl}{x}{slx}{}

\DeclareFontSeriesChangeRule {ubc}{b}{bc}{}
\DeclareFontSeriesChangeRule {ubc}{c}{ubc}{}
\DeclareFontSeriesChangeRule {ubc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {ubc}{ec}{ubec}{}
\DeclareFontSeriesChangeRule {ubc}{el}{elc}{}
\DeclareFontSeriesChangeRule {ubc}{l}{lc}{}
\DeclareFontSeriesChangeRule {ubc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {ubc}{sc}{ubsc}{}
\DeclareFontSeriesChangeRule {ubc}{sl}{slc}{}
```

```
\DeclareFontSeriesChangeRule {ubc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {ubc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {ubc}{x}{ubx}{}

\DeclareFontSeriesChangeRule {ubx}{b}{bx}{}
\DeclareFontSeriesChangeRule {ubx}{c}{ubc}{}
\DeclareFontSeriesChangeRule {ubx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {ubx}{ec}{ubec}{}
\DeclareFontSeriesChangeRule {ubx}{el}{elx}{}
\DeclareFontSeriesChangeRule {ubx}{l}{lx}{}
\DeclareFontSeriesChangeRule {ubx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {ubx}{sc}{ubsc}{}
\DeclareFontSeriesChangeRule {ubx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {ubx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {ubx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {ubx}{x}{ubx}{}

\DeclareFontSeriesChangeRule {ub}{c}{ubc}{}
\DeclareFontSeriesChangeRule {ub}{ec}{ubec}{}
\DeclareFontSeriesChangeRule {ub}{sc}{ubsc}{}
\DeclareFontSeriesChangeRule {ub}{x}{ubx}{}

\DeclareFontSeriesChangeRule {ulc}{b}{bc}{}
\DeclareFontSeriesChangeRule {ulc}{c}{ulc}{}
\DeclareFontSeriesChangeRule {ulc}{eb}{ebc}{}
\DeclareFontSeriesChangeRule {ulc}{ec}{ulec}{ulc}
\DeclareFontSeriesChangeRule {ulc}{el}{elc}{}
\DeclareFontSeriesChangeRule {ulc}{l}{lc}{}
\DeclareFontSeriesChangeRule {ulc}{sb}{sbc}{}
\DeclareFontSeriesChangeRule {ulc}{sc}{ulsc}{ulc}
\DeclareFontSeriesChangeRule {ulc}{sl}{slc}{}
\DeclareFontSeriesChangeRule {ulc}{ub}{ubc}{}
\DeclareFontSeriesChangeRule {ulc}{ul}{ulc}{}
\DeclareFontSeriesChangeRule {ulc}{x}{ulx}{}

\DeclareFontSeriesChangeRule {ulx}{b}{bx}{}
\DeclareFontSeriesChangeRule {ulx}{c}{ulc}{}
\DeclareFontSeriesChangeRule {ulx}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {ulx}{ec}{ulec}{}
\DeclareFontSeriesChangeRule {ulx}{el}{elx}{}
\DeclareFontSeriesChangeRule {ulx}{l}{lx}{}
\DeclareFontSeriesChangeRule {ulx}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {ulx}{sc}{ulsc}{}
\DeclareFontSeriesChangeRule {ulx}{sl}{slx}{}
\DeclareFontSeriesChangeRule {ulx}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {ulx}{ul}{ulx}{}
\DeclareFontSeriesChangeRule {ulx}{x}{ulx}{}

\DeclareFontSeriesChangeRule {ul}{c}{ulc}{}
\DeclareFontSeriesChangeRule {ul}{ec}{ulec}{}
\DeclareFontSeriesChangeRule {ul}{sc}{ulsc}{}
\DeclareFontSeriesChangeRule {ul}{x}{ulx}{}

\DeclareFontSeriesChangeRule {x}{b}{bx}{}
\DeclareFontSeriesChangeRule {x}{c}{c}{}
\DeclareFontSeriesChangeRule {x}{eb}{ebx}{}
\DeclareFontSeriesChangeRule {x}{ec}{ec}{}
\DeclareFontSeriesChangeRule {x}{el}{elx}{}
```

```
\DeclareFontSeriesChangeRule {x}{l}{lx}{}
\DeclareFontSeriesChangeRule {x}{sb}{sbx}{}
\DeclareFontSeriesChangeRule {x}{sc}{sc}{}
\DeclareFontSeriesChangeRule {x}{sl}{slx}{}
\DeclareFontSeriesChangeRule {x}{ub}{ubx}{}
\DeclareFontSeriesChangeRule {x}{ul}{ulx}{}
```

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

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

```
\DeclareFontSeriesChangeRule {bc}{m?}{c}{}
\DeclareFontSeriesChangeRule {bec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {bsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {bx}{m?}{x}{}
\DeclareFontSeriesChangeRule {b}{m?}{m}{}
\DeclareFontSeriesChangeRule {c}{m?}{c}{}
\DeclareFontSeriesChangeRule {ebc}{m?}{c}{}
\DeclareFontSeriesChangeRule {ebec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {ebsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {ebx}{m?}{x}{}
\DeclareFontSeriesChangeRule {eb}{m?}{m}{}
\DeclareFontSeriesChangeRule {ec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {elc}{m?}{c}{}
\DeclareFontSeriesChangeRule {elec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {elsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {elx}{m?}{x}{}
\DeclareFontSeriesChangeRule {el}{m?}{m}{}
\DeclareFontSeriesChangeRule {lc}{m?}{c}{}
\DeclareFontSeriesChangeRule {lec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {lsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {lx}{m?}{x}{}
\DeclareFontSeriesChangeRule {l}{m?}{m}{}
\DeclareFontSeriesChangeRule {m}{m?}{m}{}
\DeclareFontSeriesChangeRule {sbc}{m?}{c}{}
\DeclareFontSeriesChangeRule {sbec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {sbsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {sbx}{m?}{x}{}
\DeclareFontSeriesChangeRule {sb}{m?}{m}{}
\DeclareFontSeriesChangeRule {sc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {slc}{m?}{c}{}
\DeclareFontSeriesChangeRule {slec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {slsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {slx}{m?}{x}{}
\DeclareFontSeriesChangeRule {sl}{m?}{m}{}
\DeclareFontSeriesChangeRule {ubc}{m?}{c}{}
\DeclareFontSeriesChangeRule {ubec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {ubsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {ubx}{m?}{x}{}
\DeclareFontSeriesChangeRule {ub}{m?}{ub}{}
\DeclareFontSeriesChangeRule {ulc}{m?}{c}{}
\DeclareFontSeriesChangeRule {ulec}{m?}{ec}{}
\DeclareFontSeriesChangeRule {ulsc}{m?}{sc}{}
\DeclareFontSeriesChangeRule {ulx}{m?}{x}{}
```

```
\DeclareFontSeriesChangeRule {ul}{m?}{m}{}
\DeclareFontSeriesChangeRule {x}{m?}{x}{}
```

And there the special rules for `?m`:

```
\DeclareFontSeriesChangeRule {bc}{?m}{b}{}
\DeclareFontSeriesChangeRule {bec}{?m}{b}{}
\DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
\DeclareFontSeriesChangeRule {bsc}{?m}{b}{}
\DeclareFontSeriesChangeRule {bx}{?m}{b}{}
\DeclareFontSeriesChangeRule {b}{?m}{b}{}
\DeclareFontSeriesChangeRule {c}{?m}{m}{}
\DeclareFontSeriesChangeRule {ebc}{?m}{eb}{}
\DeclareFontSeriesChangeRule {ebec}{?m}{eb}{}
\DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
\DeclareFontSeriesChangeRule {ebsc}{?m}{eb}{}
\DeclareFontSeriesChangeRule {ebx}{?m}{eb}{}
\DeclareFontSeriesChangeRule {eb}{?m}{eb}{}
\DeclareFontSeriesChangeRule {ec}{?m}{m}{}
\DeclareFontSeriesChangeRule {elc}{?m}{el}{}
\DeclareFontSeriesChangeRule {elec}{?m}{el}{}
\DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
\DeclareFontSeriesChangeRule {elsc}{?m}{el}{}
\DeclareFontSeriesChangeRule {elx}{?m}{el}{}
\DeclareFontSeriesChangeRule {el}{?m}{el}{}
\DeclareFontSeriesChangeRule {lc}{?m}{l}{}
\DeclareFontSeriesChangeRule {lec}{?m}{l}{}
\DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
\DeclareFontSeriesChangeRule {lsc}{?m}{l}{}
\DeclareFontSeriesChangeRule {lx}{?m}{l}{}
\DeclareFontSeriesChangeRule {l}{?m}{l}{}
\DeclareFontSeriesChangeRule {m}{?m}{m}{}
\DeclareFontSeriesChangeRule {sbc}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sbec}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sbsc}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sbx}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sb}{?m}{sb}{}
\DeclareFontSeriesChangeRule {sc}{?m}{m}{}
\DeclareFontSeriesChangeRule {sc}{?m}{m}{}
\DeclareFontSeriesChangeRule {slc}{?m}{sl}{}
\DeclareFontSeriesChangeRule {slec}{?m}{sl}{}
\DeclareFontSeriesChangeRule {slsc}{?m}{sl}{}
\DeclareFontSeriesChangeRule {slsc}{?m}{sl}{}
\DeclareFontSeriesChangeRule {slx}{?m}{sl}{}
\DeclareFontSeriesChangeRule {sl}{?m}{sl}{}
\DeclareFontSeriesChangeRule {ubc}{?m}{ub}{}
\DeclareFontSeriesChangeRule {ubec}{?m}{ub}{}
\DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
\DeclareFontSeriesChangeRule {ubsc}{?m}{ub}{}
\DeclareFontSeriesChangeRule {ubx}{?m}{ub}{}
\DeclareFontSeriesChangeRule {ub}{?m}{m}{}
\DeclareFontSeriesChangeRule {ulc}{?m}{ul}{}
\DeclareFontSeriesChangeRule {ulec}{?m}{ul}{}
\DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}
\DeclareFontSeriesChangeRule {ulsc}{?m}{ul}{}
```

```
\DeclareFontSeriesChangeRule {ulx}{?m}{ul}{}
\DeclareFontSeriesChangeRule {ul}{?m}{ul}{}
\DeclareFontSeriesChangeRule {x}{?m}{m}{}
```

1.3 Changing to a new series

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

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

(*End definition for* `\if@forced@series`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\fontseriesforce`*. This function is documented on page* **??**.)

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

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

(*End definition for* `\fontseries`*. This function is documented on page* **??**.)

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

```
\def\merge@font@series#1{%
  \expandafter\expandafter\expandafter
  \merge@font@series@
    \csname series@\f@series @#1\endcsname
    {#1}%
    \@nil
}
```

(*End definition for* `\merge@font@series`*. This function is documented on page* **??**.)

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

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

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

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

```
%\ifcat\expandafter X\detokenize{#1}X%
  \def\reserved@a{#3}%
  \ifx\reserved@a\@empty
    \set@target@series{#2}%
  \else
```

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

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

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

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

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

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

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

```
          \set@target@series{#2}%
          \@font@shape@subst@warning
```

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

```
        \else
          \set@target@series{#3}%
          \@font@shape@subst@warning
        \fi
    \fi
  \fi
}
```

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

```
\def\@font@shape@subst@warning{%
   \edef\reserved@b{\curr@fontshape}%
   \ifx\reserved@a\reserved@b \else
     \@font@warning{Font shape `\reserved@a' undefined\MessageBreak
                    using `\reserved@b' instead}%
   \fi
}
```

(*End definition for* \merge@font@series@. *This function is documented on page* ??.)

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

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

(*End definition for* \maybe@load@fontshape. *This function is documented on page* ??.)

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

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

eclareFontSeriesChangeRule {m}{sc}{msc}{mc}

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

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

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

```
    \edef\f@series{#1}%
    \series@maybe@drop@one@m\f@series\f@series
}
```

(*End definition for* \set@target@series*. This function is documented on page* ??.)

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

```
\def\series@maybe@drop@one@m#1{%
  \expandafter\series@maybe@drop@one@m@x\expandafter{#1}}

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

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

```
  \def\in@@##1,#1,{}%
  \series@check@toks\expandafter{\in@@
    ,ulm,elm,lm,slm,mm,sbm,bm,ebm,ubm,muc,mec,mc,msc,msx,mx,mex,mux,{}{},#1,}%
  \edef\in@@{\the\series@check@toks}%
  \ifx\in@@\@empty
```

The default definition for \bfdefault etc is actually b\@empty so that we can detect if the user has changed the default. However that means a) the above test will definitely fail (maybe something to change) and b) we better use \edef on the next line to get rid of it as otherwise the test against #2 (e.g. \bfdef@ult) will fail in other places.

```
    \edef#2{#1}%
  \else
    \edef#2{\expandafter\series@drop@one@m #1m\series@drop@one@m}%
  \fi
}
```

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

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

(*End definition for* \series@maybe@drop@one@m*. This function is documented on page* ??.)

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

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

(*End definition for* \series@drop@one@m*. This function is documented on page* ??.)

2 Changing the shape

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

\DeclareFontShapeChangeRule

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

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

(*End definition for* `\DeclareFontShapeChangeRule`*. This function is documented on page* **??**.)

There is kind of the same problem with returning back from `sc` to normal. It sort of needs its own letter. In `fontspec` this was solved by the first time `\upshape` changes `it` or `sl` back (so only `sc` remains) and second time it changes then `sc` back to normal. Maybe that's not a bad way to handle it, but decided for a slightly different approach: `n` always returns to "normal", ie resets everything and `up` changes italic or slanted to upright and `ulc` undoes small caps.

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

\ulcshape
\textulc
\ulcdefault

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

```
\DeclareRobustCommand\ulcshape
        {\not@math@alphabet\ulcshape\relax
         \fontshape\ulcdefault\selectfont}
\let\ulcdefault\@undefined      % for rollback
\newcommand\ulcdefault{ulc}
```

(*End definition for* `\ulcshape`*,* `\textulc`*, and* `\ulcdefault`*. These functions are documented on page* **??**.)

\swshape
\textsw
\swdefault

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

```
\DeclareRobustCommand\swshape
        {\not@math@alphabet\swshape\relax
         \fontshape\swdefault\selectfont}
\let\swdefault\@undefined      % for rollback
\newcommand\swdefault{sw}
```

(*End definition for* `\swshape`*,* `\textsw`*, and* `\swdefault`*. These functions are documented on page* **??**.)

\sscshape
\textssc
\sscdefault

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

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

```
\DeclareRobustCommand\sscshape
        {\not@math@alphabet\sscshape\relax
```

```
          \fontshape\sscdefault\selectfont}
\let\sscdefault\@undefined      % for rollback
\newcommand\sscdefault{ssc}
```

(*End definition for* `\sscshape`*,* `\textssc`*, and* `\sscdefault`*. These functions are documented on page* ??.)

2.1 Mapping rules for shape combinations

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

```
%\DeclareFontShapeChangeRule {n}{n}   {n}   {}
\DeclareFontShapeChangeRule {n}{it}  {it}  {sl}
\DeclareFontShapeChangeRule {n}{sl}  {sl}  {it}
%\DeclareFontShapeChangeRule {n}{sw}  {sw}  {}
%\DeclareFontShapeChangeRule {n}{sc}  {sc}  {}
\DeclareFontShapeChangeRule {n}{ulc} {n}   {}
\DeclareFontShapeChangeRule {n}{up} {n}    {}
```

```
%\DeclareFontShapeChangeRule {it}{n}   {n}     {}
%\DeclareFontShapeChangeRule {it}{it}  {it}    {}
\DeclareFontShapeChangeRule {it}{sl}  {sl}    {it}
%\DeclareFontShapeChangeRule {it}{sw}  {sw}    {}
```

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

```
\DeclareFontShapeChangeRule {it}{sc}  {scit}  {scsl}
\DeclareFontShapeChangeRule {it}{ulc} {it}    {}
\DeclareFontShapeChangeRule {it}{up} {n}      {}
```

```
%\DeclareFontShapeChangeRule {sl}{n}   {n}     {}
\DeclareFontShapeChangeRule {sl}{it}  {it}    {sl}
%\DeclareFontShapeChangeRule {sl}{sl}  {sl}    {}
%\DeclareFontShapeChangeRule {sl}{sw}  {sw}    {}
\DeclareFontShapeChangeRule {sl}{sc}  {scsl}  {scit}
\DeclareFontShapeChangeRule {sl}{ulc} {sl}    {}
\DeclareFontShapeChangeRule {sl}{up} {n}      {}
```

```
%\DeclareFontShapeChangeRule {sc}{n}   {n}     {}
\DeclareFontShapeChangeRule {sc}{it}  {scit}  {scsl}
\DeclareFontShapeChangeRule {sc}{sl}  {scsl}  {scit}
\DeclareFontShapeChangeRule {sc}{sw}  {scsw}  {sw}
%\DeclareFontShapeChangeRule {sc}{sc}  {sc}    {}
\DeclareFontShapeChangeRule {sc}{ulc} {n}     {}
```

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

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

```
%\DeclareFontShapeChangeRule {sc}{up} {sc}     {}
\DeclareFontShapeChangeRule {sc}{up} {n}     {}
```

```
%\DeclareFontShapeChangeRule {scit}{n}   {n}    {}
\DeclareFontShapeChangeRule {scit}{it}  {scit}  {}
\DeclareFontShapeChangeRule {scit}{sl}  {scsl}  {scit}
\DeclareFontShapeChangeRule {scit}{sw}  {scsw}  {sc}    % or scit?
\DeclareFontShapeChangeRule {scit}{sc}  {scit}  {}
\DeclareFontShapeChangeRule {scit}{ulc} {it}    {}
\DeclareFontShapeChangeRule {scit}{up} {sc}     {}
```

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

```
%\DeclareFontShapeChangeRule {scsl}{n}   {n}     {}
\DeclareFontShapeChangeRule {scsl}{it}  {scit}  {scsl}
\DeclareFontShapeChangeRule {scsl}{sl}  {scsl}  {}
\DeclareFontShapeChangeRule {scsl}{sw}  {scsw}  {sc}    % or scsl?
\DeclareFontShapeChangeRule {scsl}{sc}  {scsl}  {}
\DeclareFontShapeChangeRule {scsl}{ulc} {sl}    {}
\DeclareFontShapeChangeRule {scsl}{up}   {sc}   {}
```

```
%\DeclareFontShapeChangeRule {scsw}{n}   {n}     {}
\DeclareFontShapeChangeRule {scsw}{it}  {scit}  {scsw}
\DeclareFontShapeChangeRule {scsw}{sl}  {scsl}  {}
\DeclareFontShapeChangeRule {scsw}{sw}  {scsw}  {}
\DeclareFontShapeChangeRule {scsw}{sc}  {scsw}  {}
\DeclareFontShapeChangeRule {scsw}{ulc} {sw}    {}
\DeclareFontShapeChangeRule {scsw}{up} {sc}     {}
```

```
%\DeclareFontShapeChangeRule {sw}{n}   {n}     {}
%\DeclareFontShapeChangeRule {sw}{it}  {it}    {}
%\DeclareFontShapeChangeRule {sw}{sl}  {sl}    {}
%\DeclareFontShapeChangeRule {sw}{sw}  {sw}    {}
\DeclareFontShapeChangeRule {sw}{sc}  {scsw}  {}
\DeclareFontShapeChangeRule {sw}{ulc} {sw}    {}
\DeclareFontShapeChangeRule {sw}{up} {n}      {}
```

2.2 Changing to a new shape

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

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

(*End definition for* `\fontshape`*. This function is documented on page* ??.)

`\fontshapeforce` The unconditional version:

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

(*End definition for* `\fontshapeforce`*. This function is documented on page* ??.)

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

```
\def\merge@font@shape#1{%
  \expandafter\expandafter\expandafter
  \merge@font@shape@
    \csname shape@\f@shape @#1\endcsname
```

```
    {#1}%
    \@nil
}
```

(*End definition for* `\merge@font@shape`*. This function is documented on page* **??**.)

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

```
\def\merge@font@shape@#1#2#3\@nil{%
  \def\reserved@a{#3}%
  \ifx\reserved@a\@empty
    \edef\f@shape{#2}%
  \else
    \maybe@load@fontshape
    \edef\reserved@a{\f@encoding /\f@family /\f@series/#1}%
     \ifcsname \reserved@a\endcsname
       \edef\f@shape{#1}%
    \else
       \ifcsname \f@encoding /\f@family /\f@series/#2\endcsname
         \edef\f@shape{#2}%
         \@font@shape@subst@warning
       \else
         \edef\f@shape{#3}%
         \@font@shape@subst@warning
       \fi
    \fi
  \fi
}
```

(*End definition for* `\merge@font@shape@`*. This function is documented on page* **??**.)

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

```
\protected\def\normalshape
    {\not@math@alphabet\normalshape\relax
      \fontshape\shapedefault\selectfont}%
```

(*End definition for* `\normalshape`*. This function is documented on page* **??**.)

3 Make sure we win . . .

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

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

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

```
\def\reinstall@nfss@defs{%
  \protected\def\upshape
          {\not@math@alphabet\upshape\relax
           \fontshape\updefault\selectfont}%
  \protected\def\slshape
          {\not@math@alphabet\slshape\relax
```

```
          \fontshape\sldefault\selectfont}%
  \protected\def\scshape
          {\not@math@alphabet\scshape\relax
           \fontshape\scdefault\selectfont}%
  \protected\def\itshape
          {\not@math@alphabet\itshape\mathit
           \fontshape\itdefault\selectfont}%
  \protected\def\ulcshape
          {\not@math@alphabet\ulcshape\relax
           \fontshape{ulc}\selectfont}%
  \protected\def\swshape
          {\not@math@alphabet\swshape\relax
           \fontshape\swdefault\selectfont}%
  \protected\def\sscshape
          {\not@math@alphabet\sscshape\relax
           \fontshape\sscdefault\selectfont}%
}
```

(*End definition for* `\reinstall@nfss@defs`. *This function is documented on page* **??**.)

Supporting rollback ...

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩   {\DeclareFontSeriesChangeRule}{Series change rules}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\fontseries[1]{\edef\f@series{#1}}
⟨latexrelease⟩\DeclareRobustCommand\fontshape [1]{\edef\f@shape{#1}}
⟨latexrelease⟩\let\fontseriesforce\@undefined
⟨latexrelease⟩\let\fontshapeforce\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\DeclareFontSeriesChangeRule\@undefined
⟨latexrelease⟩\let\merge@font@series\@undefined
⟨latexrelease⟩\let\merge@font@series@\@undefined
⟨latexrelease⟩\let\@font@shape@subst@warning\@undefined
⟨latexrelease⟩\let\maybe@load@fontshape\@undefined
⟨latexrelease⟩\let\set@target@series\@undefined
⟨latexrelease⟩\let\series@maybe@drop@one@m\@undefined
⟨latexrelease⟩\let\series@drop@one@m\@undefined
⟨latexrelease⟩\let\DeclareFontShapeChangeRule\@undefined
⟨latexrelease⟩\let\merge@font@shape\@undefined
⟨latexrelease⟩\let\merge@font@shape@\@undefined
⟨latexrelease⟩\let\normalshape\@undefined
⟨latexrelease⟩\let\ulcshape\@undefined
⟨latexrelease⟩\let\ulcdefault\@undefined
⟨latexrelease⟩\let\swshape\@undefined
⟨latexrelease⟩\let\swdefault\@undefined
⟨latexrelease⟩\let\sscshape\@undefined
⟨latexrelease⟩\let\sscdefault\@undefined
⟨latexrelease⟩\let\normalshape\@undefined
```

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

```
⟨latexrelease⟩
⟨latexrelease⟩\let\reinstall@nfss@defs\relax
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

This initializes the 2020/02/02 extensions to NFSS after any changes in the preamble.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\reinstall@nfss@defs}{NFSS series init}%
\g@addto@macro\@kernel@after@begindocument@before
               {\reinstall@nfss@defs\init@series@setup}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

The initialization was introduced in 2020/02/02 but

```
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\reinstall@nfss@defs}{NFSS series init}%
⟨latexrelease⟩\AtBeginDocument{\reinstall@nfss@defs\init@series@setup}
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\reinstall@nfss@defs}{NFSS series init}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
⟨/2ekernel⟩
```

File t
ltfsstrc.dtx

1 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.

2 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⟩
```

3 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`.[11]

```
⟨+debug⟩ \input trace.sty
```

4 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⟩
\def\tracingfonts{%
  \@font@warning{Command \noexpand\tracingfonts
           not provided.\MessageBreak
           Use the ‘tracefnt’ package.\MessageBreak Command found:}%
       \count@}
⟨/2ekernel⟩
```

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⟩
```

(*End definition for* `\tracingfonts`. *This function is documented on page* ??.)

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}}%
```

[11] 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{%
         \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}}%
    \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⟩
\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⟩
```

5 Macros common to fam.tex and tracefnt.sty

In the first versions of tracefnt.dtx some macros of fam.dtx[12] 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.

5.1 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⟩
\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
```

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⟩
```

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.

(*End definition for* \extract@font. *This function is documented on page* ??.)

\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⟩
\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
```

[12] This file is currently not distributed in documented form. Its code is part of ltfss.dtx.

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⟩
```

(*End definition for* `\get@external@font`. *This function is documented on page* **??**.)

`\selectfont` The macro `\selectfont` is called whenever a font change must take place.

```
⟨*2ekernel | package⟩
\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
    }
```

(*End definition for* \selectfont. *This function is documented on page* ??.)

\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!

(*End definition for* \set@fontsize. *This function is documented on page* ??.)

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

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

(*End definition for* \size@update. *This function is documented on page* ??.)

\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⟩
```

(*End definition for* \type@restoreinfo. *This function is documented on page* ??.)

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

```
\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⟩
```

(*End definition for* `\glb@settings` *and* `\glb@currsize`*. These functions are documented on page* **??**.)

`\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⟩
\def\baselinestretch{1}
```

(*End definition for* `\baselinestretch`*. This function is documented on page* **??**.)

`\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⟩
```

(*End definition for* `\every@math@size`*. This function is documented on page* **??**.)

5.2 Math fonts setup

5.2.1 Outline of algorithm for math font sizes

TEX uses the math fonts that are current when the end of a formula is reached. If we don't want to keep font setups local to every formula (which would result in an enormous overhead, we have to be careful not to end up with the wrong setup in case formulas are nested, e.g., we need to be able to handle

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

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

This is handled in the following way:

1. At any point in the document the global variable `\gbl@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).

5.2.2 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⟩
\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}%
}
```

(*End definition for* \check@mathfonts*. This function is documented on page* ??.)

\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⟩
```

(*End definition for* \init@restore@glb@settings*. This function is documented on page* ??.)

\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
}
```

(*End definition for* `\restglb@settings`*. This function is documented on page* **??**.)

5.2.3 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 $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX macros for placing accents.

(*End definition for* `\use@mathgroup`*. This function is documented on page* **??**.)

\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⟩
```

(*End definition for* \math@egroup. *This function is documented on page* ??.)

\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.[13] 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
```

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}%
```

[13] 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.

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

```
⟨+debug⟩ \poptracing
    }
⟨/2ekernel | package⟩
```

(*End definition for* `\getanddefine@fonts`. *This function is documented on page* ??.)

6 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⟩
\def\ifnot@nil#1{\def\reserved@a{#1}%
  \ifx\reserved@a\@nnil \expandafter\@gobble
  \else \expandafter\@firstofone\fi}
```

(*End definition for* `\ifnot@nil`. *This function is documented on page* ??.)

`\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}
```

(*End definition for* `\remove@to@nnil`, `\remove@angles`, *and* `\remove@star`. *These functions are documented on page* ??.)

`\extract@sizefn` This macro takes a size specification and parses it into size function and the optional and mandatory arguments.

```
\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
  }
```

(*End definition for* `\extract@sizefn`. *This function is documented on page* ??.)

`\try@simple@size` This function tries to extract the given size (specified by `\f@size`) for the requested font shape. The font information must already be present in `\font@info`. The central macro that does the real work is `\extract@fontinfo`. We will first give a simple example how this macro works, and describe it in full generality later.

Assume that the requested parameters are: *encoding scheme* 'OT1', *family* 'cm', *series* 'sansserif', *shape* 'normal', and *size* '12'. The corresponding font definitions have already been extracted from the macro `\OT1/cm/sansserif/normal` and stored in `font@info`. (Otherwise `\extract@fontinfo` doesn't get called.) This information consists of a token list made of characters of category code 12 of the form

```
<10*>cmss10<12*>cmss12<17*>cmss17
```

For reasonable packages one usually needs more sizes but this is sufficient to get the flavour. We will define a macro `\extract@fontinfo` to find the external font name ('cmss12') for us:

```
\def\extract@fontinfo#1<12*#2>#3<#4\@nnil{%
    \set@simple@size@args#3<#4\@nnil
    \execute@size@function{#2}}
```

so that when it gets called via

```
\extract@fontinfo<10*>cmss10<12*>cmss12<17*>cmss17\@nnil
```

`#1` will contain all characters before `<12*>`, `#2` will be empty, `#3` will be exactly `cmss12`, and `#3` will be `17>cmss17`. The expansion is therefore

```
\set@simple@size@args cmss12<17*>cmss17\@nnil
\execute@size@function{}
```

This means: the default (empty) size function will be executed, with its optional argument 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
}
```

(*End definition for* \try@simple@size*. This function is documented on page* ??.)

\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}
```

(*End definition for* \set@simple@size@args*. This function is documented on page* ??.)

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>}
```

(*End definition for* \extract@rangefontinfo*. This function is documented on page* ??.)

\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}
```

(*End definition for* \is@range*. This function is documented on page* ??.)

\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}}
```

(*End definition for* `\check@range`*. This function is documented on page* **??**.)

`\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.

```
\newdimen\lower@bound
\newdimen\upper@bound
```

(*End definition for* `\lower@bound` *and* `\upper@bound`*. These functions are documented on page* **??**.)

`\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).

```
\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 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}
```

(*End definition for* `\check@single`*. This function is documented on page* **??**.)

`\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⟩

(*End definition for* `\set@size@funct@args` *and* `\set@size@funct@args@`*. These functions are documented on page* **??**.)

`\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⟩
\def\DeclareSizeFunction#1#2{\@namedef{s@fct@#1}{#2}}
\@onlypreamble\DeclareSizeFunction
⟨/2ekernel⟩
```

(*End definition for* `\DeclareSizeFunction`*. This function is documented on page* **??**.)

`\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⟩
\def\execute@size@function#1{%
⟨*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⟩
```

(*End definition for* `\execute@size@function`*. This function is documented on page* **??**.)

`\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⟩
\def\try@size@range{%
    \expandafter\extract@rangefontinfo\font@info <-*>\@nil<\@nnil
}
```

(*End definition for* `\try@size@range`*. This function is documented on page* **??**.)

`\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.

```
\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
}
```

(*End definition for* `\try@size@substitution`*. This function is documented on page* **??**.)

\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{document}. The macro \fontsubfuzz contains the amount that will not cause terminal warnings (warnings still go into the transcript file).

```
\def\font@submax{0pt}
\def\fontsubfuzz{.4pt}
⟨/2ekernel⟩
⟨+package⟩\def\fontsubfuzz{0pt}
```

(*End definition for* \font@submax *and* \fontsubfuzz*. These functions are documented on page* ??.)

\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⟩
\gdef\try@simples#1<#2>{%
  \tryif@simple#2->\tryif@simple}
```

(*End definition for* \try@simples*. This function is documented on page* ??.)

\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.

```
\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
```

This macro substitutes the size recorded in `\best@size` for the unavailable size `\f@size`.
`\subst@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}
```

(*End definition for* `\tryis@simple` *and* `\subst@size`*. These functions are documented on page* ??.)

6.1 Sizefunctions

In the following we define some useful size functions.

`\s@fct@` This is the default size function. Mandatory argument is an external font name, optional argument a scale factor. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```
\DeclareSizeFunction{}{\empty@sfcnt\@font@warning}
\DeclareSizeFunction{s}{\empty@sfcnt\@font@info}
```

```
\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}}
```

(*End definition for* `\s@fct@`. *This function is documented on page* **??**.)

`\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 optional argument a scale factor. The font is scaled to `\f@size` if no optional argument is present, and to `\f@size` multiplied by the optional argument otherwise.

```
\DeclareSizeFunction{gen}{\gen@sfcnt\@font@warning}
\DeclareSizeFunction{sgen}{\gen@sfcnt\@font@info}
```

```
\def\gen@sfcnt{%
      \edef\mandatory@arg{\mandatory@arg\f@size}%
      \empty@sfcnt}
```

(*End definition for* `\s@fct@gen` *and* `\s@fct@sgen`. *These functions are documented on page* **??**.)

`\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.

```
\DeclareSizeFunction{genb}{\genb@sfcnt\@font@warning}
\DeclareSizeFunction{sgenb}{\genb@sfcnt\@font@info}
```

```
\def\genb@sfcnt{%
    \edef\mandatory@arg{\mandatory@arg\expandafter\genb@x\f@size..\@@}%
    \empty@sfcnt}
```

(*End definition for* `\s@fct@genb` *and* `\s@fct@sgenb`. *These functions are documented on page* **??**.)

`\genb@x` `\genb@y`

The auxiliary macros `\genb@x` and `\genb@y` are used to convert the `\f@size` into centipoints.

```
\def\genb@x#1.#2.#3\@@{\two@digits{#1}\genb@y#200\@@}
\def\genb@y#1#2#3\@@{#1#2}
```

(*End definition for* `\genb@x` *and* `\genb@y`. *These functions are documented on page* **??**.)

`\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.

```
\DeclareSizeFunction{sub}{\sub@sfcnt\@font@warning}
\DeclareSizeFunction{ssub}{\sub@sfcnt\@font@info}
```

```
\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
}
```

(*End definition for* `\s@fct@sub`. *This function is documented on page* ??.)

`\@font@aliasinfo` Sometimes a substitution is only done to map a long font name to a standard shape or series, e.g.,

```
DeclareFontShape{T1}{Roboto-LF}{b}{it}{<-> alias * Roboto-LF/bold/it}{}
```

Using the `ssub` function in that case will give a strange (and incorrect) warning. As an alternative we therefore offer the size function `alias`. It will still add some info into the `.log` file, but no longer complains that the font shape is not available. It is implemented by grabbing the default warning text and replacing it with a new one.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\@font@aliasinfo}{alias size function}%
\DeclareSizeFunction{alias}{\sub@sfcnt\@font@aliasinfo}
\def\@font@aliasinfo#1{%
  \@font@info{Font\space shape\space `\curr@fontshape'\space
              aliased\space to\MessageBreak `\mandatory@arg'}%
}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@font@aliasinfo}{alias size function}%
⟨latexrelease⟩\let\s@fct@alias\@undefined
⟨latexrelease⟩\let\@font@aliasinfo\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@font@aliasinfo`*. This function is documented on page* **??**.)

`\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.

```
\DeclareSizeFunction{subf}{\subf@sfcnt\@font@warning}
\DeclareSizeFunction{ssubf}{\subf@sfcnt\@font@info}
```

```
\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%
      }
```

(*End definition for* `\s@fct@subf`*. This function is documented on page* **??**.)

`\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.

```
\DeclareSizeFunction{fixed}{\fixed@sfcnt\@font@warning}
\DeclareSizeFunction{sfixed}{\fixed@sfcnt\@font@info}
```

```
\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⟩
```

(*End definition for* `\s@fct@fixed`*. This function is documented on page* **??**.)

File u
ltfsscmp.dtx

This file contains the implementation of commands giving compatibility with the original 'NFSS1' release of the Font Selection Scheme.

> **Warning:** The macro documentation is still basically the documentation from the first NFSS release and therefore in some cases probably not completely accurate.

Version 1 of NFSS is obsolete now for about 20 years (and was "current" only for a short intermediate time) so with the 2015 release these internal interface commands are removed from the kernel and made available via latexrelease package so that backward compatibility remains ensured for very old documents.

```
⟨*latexrelease⟩
\IncludeInRelease{2015/01/01}{\new@fontshape}%
                             {NFSS version1 commands}%
\let\new@fontshape\@undefined
\let\warn@rel@i\@undefined
\let\scan@fontshape\@undefined
\let\scan@@fontshape\@undefined
\let\subst@fontshape\@undefined
\let\extra@def\@undefined
\let\default@mextra\@undefined
\let\preload@sizes\@undefined
\let\err@rel@i\@undefined
\let\newmathalphabet\@undefined
\let\newmathalphabet@\@undefined
\let\newmathalphabet@@@\@undefined
\let\if@no@font@opt\@undefined
\let\@no@font@optfalse\@undefined
\let\define@mathalphabet\@undefined
\let\define@mathgroup\@undefined
\let\addtoversion\@undefined
\EndIncludeInRelease
```

In older releases we provide the original definitions.

```
\IncludeInRelease{0000/00/00}{\new@fontshape}%
                             {NFSS version1 commands}%
```

`\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
```

(*End definition for* `\new@fontshape`. *This function is documented on page* **??**.)

`\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
```

(*End definition for* `\warn@rel@i`*. This function is documented on page* ??.)

`\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
```

(*End definition for* `\scan@fontshape`*. This function is documented on page* ??.)

`\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
```

(*End definition for* `\scan@@fontshape`. *This function is documented on page* ??.)

`\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
```

(*End definition for* `\subst@fontshape`. *This function is documented on page* ??.)

`\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
```

(*End definition for* `\extra@def`. *This function is documented on page* ??.)

`\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
```

(*End definition for* `\default@mextra`. *This function is documented on page* ??.)

`\preload@sizes` The new interface is `\DeclarePreloadSizes`.

```
\gdef\preload@sizes{%
     \warn@rel@i\preload@sizes\DeclarePreloadSizes
     \DeclarePreloadSizes U%
}%
\@onlypreamble\preload@sizes
```

(*End definition for* `\preload@sizes`. *This function is documented on page* ??.)

`\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
```

(*End definition for* `\err@rel@i`*. This function is documented on page* **??**.)

`\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@@@
```

(*End definition for* `\newmathalphabet`*,* `\newmathalphabet@@`*, and* `\newmathalphabet@@@`*. These functions are documented on page* **??**.)

`\if@no@font@opt`
`\@no@font@optfalse`

```
\global\let\if@no@font@opt\iftrue
\gdef\@no@font@optfalse{\let\if@no@font@opt\iffalse}%
```

(*End definition for* `\if@no@font@opt` *and* `\@no@font@optfalse`*. These functions are documented on page* **??**.)

`\define@mathalphabet`

This is a case where dying is best.

```
\gdef\define@mathalphabet{%
      \err@rel@i\define@mathalphabet\DeclareMathAlphabet
}%
\@onlypreamble\define@mathalphabet
```

(*End definition for* `\define@mathalphabet`*. This function is documented on page* **??**.)

`\define@mathgroup`

And here is another one

```
\gdef\define@mathgroup{%
      \err@rel@i\define@mathgroup\DeclareSymbolFont
}%
\@onlypreamble\define@mathgroup
```

(*End definition for* `\define@mathgroup`*. This function is documented on page* **??**.)

\addtoversion \addtoversion is the old form.

```
\def\addtoversion#1#2{%
  \warn@rel@i\addtoversion\SetMathAlphabet
  \SetMathAlphabet#2{#1}{U}}%
\@onlypreamble\addtoversion
```

(*End definition for* \addtoversion*. This function is documented on page* **??**.)

Finishing off this huge \IncludeInRelease argument:

```
\EndIncludeInRelease
⟨/latexrelease⟩
```

File v
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.

1 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⟩
\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@
```

(*End definition for* `\in@` *and* `\ifin@`*. These functions are documented on page* **??**.)

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`⟨*version*$_1$⟩`\version@elt`⟨*version*$_2$⟩... `\version@elt`⟨*version*$_n$⟩

- 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
```

(*End definition for* \version@list*. This function is documented on page* ??.)

\version@elt

```
\let\version@elt\relax
\@onlypreamble\version@elt
```

(*End definition for* \version@elt*. This function is documented on page* ??.)

\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}
```

(*End definition for* `\new@mathversion`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\alpha@list`*. This function is documented on page* **??**.)

`\alpha@elt`

```
\let\alpha@elt\relax
\@onlypreamble\alpha@elt
```

(*End definition for* `\alpha@elt`*. This function is documented on page* **??**.)

`\newgroup` Start the group (fam) allocation at 0. (Doesn't belong here.)

```
\count18=-1
```

(*End definition for* `\newgroup`*. This function is documented on page* **??**.)

`\stepcounter`

(*End definition for* `\stepcounter`*. This function is documented on page* **??**.)

`\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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}
⟨latexrelease⟩                  {\select@group}{\select@group}%
⟨*2ekernel | latexrelease⟩
\def\select@group#1#2#3#4{%
 \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
 {%
 \ifmmode
  \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
     \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}%
 }%
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}
⟨latexrelease⟩                  {\select@group}{\select@group}%
⟨latexrelease⟩\def\select@group#1#2#3#4{%
⟨latexrelease⟩ \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
⟨latexrelease⟩ {%
⟨latexrelease⟩ \ifmmode
⟨latexrelease⟩  \ifnum\csname c@mv@\math@version\endcsname<\sixt@@n
⟨latexrelease⟩     \begingroup
⟨latexrelease⟩       \escapechar\m@ne
⟨latexrelease⟩       \getanddefine@fonts
⟨latexrelease⟩         {\csname c@mv@\math@version\endcsname}#3%
⟨latexrelease⟩       \globaldefs\@ne  \math@fonts
⟨latexrelease⟩     \endgroup
⟨latexrelease⟩     \init@restore@version
⟨latexrelease⟩     \xdef#1{\noexpand\use@mathgroup\noexpand#2%
⟨latexrelease⟩             {\number\csname c@mv@\math@version\endcsname}}%
⟨latexrelease⟩     \global\advance\csname c@mv@\math@version\endcsname\@ne
⟨latexrelease⟩   \else
⟨latexrelease⟩     \let#1\relax
⟨latexrelease⟩     \@latex@error{Too many math alphabets used in
⟨latexrelease⟩                   version \math@version}%
```

```
⟨latexrelease⟩         \@eha
⟨latexrelease⟩    \fi
⟨latexrelease⟩ \else \expandafter\non@alpherr\fi
⟨latexrelease⟩ #1{#4}%
⟨latexrelease⟩ }%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\restore@mathversion
```

(*End definition for* `\select@group`. *This function is documented on page* **??**.)

`\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
```

(*End definition for* `\init@restore@version`. *This function is documented on page* **??**.)

`\non@alpherr`

```
\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}
```

(*End definition for* `\non@alpherr`. *This function is documented on page* **??**.)

`\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
```

(*End definition for* `\dorestore@version`. *This function is documented on page* **??**.)

`\document@select@group` We surround `\select@group` with braces so that functions using it can be used directly after `_` or `^`.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}
⟨latexrelease⟩  {\document@select@group}{\document@select@group}%
⟨*2ekernel | latexrelease⟩
\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<\e@mathgroup@top
     \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
```

If the legacy interface is used, e.g., `$\sf -1$` the math alphabet `#1` does not take an argument so we better do not surround `#4` with braces, because then we get `{\relax}` into the formula and introduce an extra Ord atom. The two different cases can be distinguished by looking at the current value of `\math@bgroup`.

```
 \expandafter#1\ifx\math@bgroup\bgroup{#4}\else#4\fi
 }%
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}
⟨latexrelease⟩  {\document@select@group}{\document@select@group}%
⟨latexrelease⟩
⟨latexrelease⟩\def\document@select@group#1#2#3#4{%
⟨latexrelease⟩ \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
⟨latexrelease⟩ {%
⟨latexrelease⟩ \ifmmode
⟨latexrelease⟩   \ifnum\csname c@mv@\math@version\endcsname<\e@mathgroup@top
⟨latexrelease⟩     \begingroup
⟨latexrelease⟩       \escapechar\m@ne
⟨latexrelease⟩       \getanddefine@fonts{\csname c@mv@\math@version\endcsname}#3%
⟨latexrelease⟩       \globaldefs\@ne  \math@fonts
⟨latexrelease⟩     \endgroup
⟨latexrelease⟩     \expandafter\extract@alph@from@version
⟨latexrelease⟩         \csname mv@\math@version\expandafter\endcsname
```

```
⟨latexrelease⟩         \expandafter{\number\csname
⟨latexrelease⟩                       c@mv@\math@version\endcsname}%
⟨latexrelease⟩          #1%
⟨latexrelease⟩     \global\advance\csname c@mv@\math@version\endcsname\@ne
⟨latexrelease⟩   \else
⟨latexrelease⟩     \let#1\relax
⟨latexrelease⟩     \@latex@error{Too many math alphabets used
⟨latexrelease⟩                   in version \math@version}%
⟨latexrelease⟩        \@eha
⟨latexrelease⟩  \fi
⟨latexrelease⟩ \else \expandafter\non@alpherr\fi
⟨latexrelease⟩ #1{#4}%
⟨latexrelease⟩ }%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}
⟨latexrelease⟩  {\document@select@group}{\document@select@group}%
⟨latexrelease⟩
⟨latexrelease⟩\def\document@select@group#1#2#3#4{%
⟨latexrelease⟩ \ifx\math@bgroup\bgroup\else\relax\expandafter\@firstofone\fi
⟨latexrelease⟩ {%
⟨latexrelease⟩ \ifmmode
⟨latexrelease⟩   \ifnum\csname c@mv@\math@version\endcsname<\sixt@@n
⟨latexrelease⟩     \begingroup
⟨latexrelease⟩       \escapechar\m@ne
⟨latexrelease⟩       \getanddefine@fonts
⟨latexrelease⟩         {\csname c@mv@\math@version\endcsname}#3%
⟨latexrelease⟩       \globaldefs\@ne  \math@fonts
⟨latexrelease⟩     \endgroup
⟨latexrelease⟩     \expandafter\extract@alph@from@version
⟨latexrelease⟩         \csname mv@\math@version\expandafter\endcsname
⟨latexrelease⟩         \expandafter{\number\csname
⟨latexrelease⟩                       c@mv@\math@version\endcsname}%
⟨latexrelease⟩          #1%
⟨latexrelease⟩     \global\advance\csname c@mv@\math@version\endcsname\@ne
⟨latexrelease⟩   \else
⟨latexrelease⟩     \let#1\relax
⟨latexrelease⟩     \@latex@error{Too many math alphabets used
⟨latexrelease⟩                   in version \math@version}%
⟨latexrelease⟩        \@eha
⟨latexrelease⟩  \fi
⟨latexrelease⟩ \else \expandafter\non@alpherr\fi
⟨latexrelease⟩ #1{#4}%
⟨latexrelease⟩ }%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\document@select@group`. *This function is documented on page* **??**.)

`\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 as they are currently pointing to error font face. We can speed up the process by just using `\edef`, thereby avoiding all kind of extra processing. Don't use `\reset@font` since that would trigger `\selectfont`.

```
    \fontencoding\encodingdefault
    \edef\f@family{\familydefault}%
    \edef\f@series{\seriesdefault}%
    \edef\f@shape{\shapedefault}%
```

Drop stuff not longer needed. We need to add many more!!!!!!

```
 \everyjob{}%
}
\@onlypreamble\process@table
```

(*End definition for* `\process@table`. *This function is documented on page* **??**.)

```
%\@onlypreamble\set@mathradical
```

`\DeclareMathVersion`

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

(*End definition for* `\DeclareMathVersion`. *This function is documented on page* **??**.)

`\new@mathversion`

```
\def\new@mathversion#1{%
  \expandafter\in@\expandafter#1\expandafter{\version@list}%
  \ifin@
    \@font@info{Redeclaring math version
                `\expandafter\@gobblefour\string#1'}%
  \else
    \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 distinguish 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
```

(*End definition for* `\new@mathversion`. *This function is documented on page* **??**.)

`\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}{%
      \ifnum\count18<15 %
        \expandafter\new@mathgroup\csname sym#1\endcsname
        \expandafter\new@symbolfont\csname sym#1\endcsname
                         {#2}{#3}{#4}{#5}%
      \else
         \@latex@error{Too many symbol fonts declared}\@eha
      \fi
     }%
     {%
      \@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
```

(*End definition for* `\DeclareSymbolFont`. *This function is documented on page* **??**.)

\group@list

```
\let\group@list\@empty
\@onlypreamble\group@list
```

(*End definition for* `\group@list`. *This function is documented on page* **??**.)

\group@elt

```
\let\group@elt\relax
\@onlypreamble\group@elt
```

(*End definition for* `\group@elt`. *This function is documented on page* **??**.)

\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
```

(*End definition for* `\new@symbolfont`. *This function is documented on page* **??**.)

\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
```

(*End definition for* `\SetSymbolFont`. *This function is documented on page* **??**.)

\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 misspelled the name of the math
       version.^^JOr you have to specify an additional package.}%
  \fi
}
\@onlypreamble\SetSymbolFont@
```

(*End definition for* `\SetSymbolFont@`*. This function is documented on page* **??**.)

`\get@cdp`

```
\def\get@cdp#1#2/#3\@nil#4{\def#4{#2}}
\@onlypreamble\get@cdp
```

(*End definition for* `\get@cdp`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\DeclareMathAlphabet`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\new@mathalphabet`. *This function is documented on page* **??**.)

`\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
```

(*End definition for* `\SetMathAlphabet`*. This function is documented on page* **??**.)

`\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 misspelled the name of the math
       version.^^JOr you have to specify an additional package.}%
  \fi
}
\@onlypreamble\SetMathAlphabet@
```

(*End definition for* `\SetMathAlphabet@`*. This function is documented on page* **??**.)

`\DeclareMathAccent` Could do with more checks like allowing single number in `#4` lowercase in `#4` etc

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                    {DeclareMathAccent}{Make math accents robust}%
\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@
```

```
          {\expandafter\@gobble\string\mathaccent}
          {\expandafter\meaning
           \csname\expandafter\@gobble\string#1\space\endcsname}}%
        \reserved@a
        \ifin@
          \expandafter\let
            \csname\expandafter\@gobble\string#1\space\endcsname
            \@undefined
          \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
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {DeclareMathAccent}{Make math accents robust}%
⟨latexrelease⟩\def\DeclareMathAccent#1#2#3#4{%
⟨latexrelease⟩  \expandafter\in@\csname sym#3\expandafter\endcsname
⟨latexrelease⟩     \expandafter{\group@list}%
⟨latexrelease⟩  \ifin@
⟨latexrelease⟩    \begingroup
⟨latexrelease⟩      \count\z@=#4\relax
⟨latexrelease⟩      \count\tw@\count\z@
⟨latexrelease⟩      \divide\count\z@\sixt@@n
⟨latexrelease⟩      \count@\count\z@
⟨latexrelease⟩      \multiply\count@\sixt@@n
⟨latexrelease⟩      \advance\count\tw@-\count@
⟨latexrelease⟩      \if\relax\noexpand#1% is command?
⟨latexrelease⟩        \edef\reserved@a{\noexpand\in@
⟨latexrelease⟩           {\expandafter\@gobble\string\mathaccent}{\meaning#1}}%
⟨latexrelease⟩        \reserved@a
⟨latexrelease⟩        \ifin@
⟨latexrelease⟩          \expandafter\set@mathaccent
⟨latexrelease⟩             \csname sym#3\endcsname#1#2%
⟨latexrelease⟩             {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
```

```
⟨latexrelease⟩          \@font@info{Redeclaring math accent \string#1}%
⟨latexrelease⟩        \else
⟨latexrelease⟩          \expandafter\ifx
⟨latexrelease⟩          \csname\expandafter\@gobble\string#1\endcsname
⟨latexrelease⟩          \relax
⟨latexrelease⟩            \expandafter\set@mathaccent
⟨latexrelease⟩               \csname sym#3\endcsname#1#2%
⟨latexrelease⟩               {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}%
⟨latexrelease⟩          \else
⟨latexrelease⟩            \@latex@error{Command ‘\string#1’ already defined}\@eha
⟨latexrelease⟩          \fi
⟨latexrelease⟩        \fi
⟨latexrelease⟩      \else
⟨latexrelease⟩       \@latex@error{Not a command name: ‘\noexpand#1’}\@eha
⟨latexrelease⟩      \fi
⟨latexrelease⟩    \endgroup
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@latex@error{Symbol font ‘#3’ is not defined}\@eha
⟨latexrelease⟩  \fi
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\DeclareMathAccent
```

(*End definition for* `\DeclareMathAccent`*. This function is documented on page* **??**.)

`\set@mathaccent`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\set@mathaccent}{makemath accents robust}%
\def\set@mathaccent#1#2#3#4{%
  \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}%
  \MakeRobust#2%
}
\@onlypreamble\set@mathaccent
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\set@mathaccent}{makemath accents robust}%
⟨latexrelease⟩
⟨latexrelease⟩\def\set@mathaccent#1#2#3#4{%
⟨latexrelease⟩  \xdef#2{\mathaccent"\mathchar@type#3\hexnumber@#1#4\relax}}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\set@mathaccent`*. This function is documented on page* **??**.)

`\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?
```

Store the command name with a space attached inside `\reserved@@b` in case we look at a robust definition.

```
        \edef\reserved@b{\expandafter\noexpand
                          \csname\expandafter\@gobble\string#1\space\endcsname}%
```

Test both `#1` and `#1␣` for containing `mathchar`.

```
        \edef\reserved@a
          {\noexpand\in@{\expandafter\@gobble\string\mathchar}%
                         {\meaning#1\expandafter\meaning\reserved@b}}%
        \reserved@a
```

Drop `#1␣` in case it was defined before.

```
        \global\expandafter\let\reserved@b\@undefined
        \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
```

(*End definition for* `\DeclareMathSymbol`*. This function is documented on page* **??**.)

`\set@mathchar`

```
\def\set@mathchar#1#2#3#4{%
  \global\mathcode`#2="\mathchar@type#3\hexnumber@#1#4\relax}
\@onlypreamble\set@mathchar
```

(*End definition for* `\set@mathchar`*. This function is documented on page* **??**.)

`\set@mathsymbol`

```
\def\set@mathsymbol#1#2#3#4{%
  \global\mathchardef#2"\mathchar@type#3\hexnumber@#1#4\relax}
\@onlypreamble\set@mathsymbol
```

(*End definition for* `\set@mathsymbol`*. This function is documented on page* **??**.)

```
%\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
```

(*End definition for* `\DeclareMathDelimiter`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@xxDeclareMathDelimiter`*. This function is documented on page* **??**.)

`\@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@
            {\expandafter\@gobble\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
```

(*End definition for* `\@DeclareMathDelimiter`. *This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@xDeclareMathDelimiter`*. This function is documented on page* **??**.)

`\set@mathdelimiter` We have to end the definition of a math delimiter like `\lfloor` with a space and not with `\relax` as we did before, because otherwise constructs involving `\abovewithdelims` will prematurely end (pr/1329)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\set@mathdelimiter}{make delimiters robust}%
\def\set@mathdelimiter#1#2#3#4#5#6{%
```

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

```
  \protected
  \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
    \hexnumber@#2#6 }%
%  \MakeRobust#3%
}
\@onlypreamble\set@mathdelimiter
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\set@mathdelimiter}{make delimiters robust}%
⟨latexrelease⟩
⟨latexrelease⟩\def\set@mathdelimiter#1#2#3#4#5#6{%
⟨latexrelease⟩  \xdef#3{\delimiter"\mathchar@type#4\hexnumber@#1#5%
⟨latexrelease⟩    \hexnumber@#2#6 }}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\set@mathdelimiter`*. This function is documented on page* **??**.)

`\set@@mathdelimiter`

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

(*End definition for* `\set@@mathdelimiter`*. This function is documented on page* **??**.)

`\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@
       {\expandafter\@gobble\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
```

(*End definition for* `\DeclareMathRadical`*. This function is documented on page* **??**.)

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
```

(*End definition for* `\mathalpha`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\mathchar@type`*. This function is documented on page* **??**.)

`\DeclareSymbolFontAlphabet`

```
\def\DeclareSymbolFontAlphabet#1#2{%
   \expandafter\DeclareSymbolFontAlphabet@
     \csname \expandafter\@gobble\string#1\space\endcsname{#2}#1}
\@onlypreamble\DeclareSymbolFontAlphabet
```

(*End definition for* `\DeclareSymbolFontAlphabet`*. This function is documented on page* **??**.)

`\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⟩
```

(*End definition for* `\DeclareSymbolFontAlphabet@`*. This function is documented on page* **??**.)

File w
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.

1 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`.

```
⟨*2ekernel⟩
```

1.1 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.

(Actually most are now defined further down in the file.)

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}
```

2 Custom series settings for main document families

This section was introduced 2020/02/02 and for now we support a full rollback (may need splitting later).

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                    {\DeclareFontSeriesDefault}{Custom series}%
```

One problem with the NFSS approach of handling the series axis turned out to be that (especially with respect to "boldness") different font families implemented different strategies. For example, with Computer Modern fonts you normally only have `bx` whereas most PostScript fonts offered only `b` but not `bx`. As a result LaTeX's standard setting for `\bfdefault` didn't work with such fonts, but if it got changed to produce `b`, then that didn't work with Computer Modern if the fonts got combined (e.g., using Computer Modern Typewriter with such fonts).

The solution back then was to provide substitution rules in the font `.fd` such that if a `bx` series got requested the `b` series got used. While this works in that particular case, it isn't a very general solution. For example, if you happen to have a font family that has several weights you may want to typeset the whole document in a somewhat lighter or darker font but if you then modify `\mddefault` to allow for this, then of course your change only works with that particular family but not with the typewriter or sans serif family you also want to use.

A better solution was provided by the `mweights` package by Bob Tennent that offers defaults on the level of the three main font families in the document: for "rm", "sf" and "tt" so that font packages could define defaults for the sans serif document font by providing `\bfseries@sf` which then was used when `\bfseries` got executed and the current family was the `\sffamily`.

`\DeclareFontSeriesDefault`

We now support this concept directly from within LaTeX and for use in font packages (or the document preamble) we offer `\DeclareFontSeriesDefault`. This declaration takes three arguments:

document family interface: Can either be `rm`, `sf` or `tt`. This is optional and if not given the overall default.

document series interface: Can be `md` or `bf`.

series value: This is the value that is going to be used with the combination is requested.

For example, `\DeclareFontSeriesDefault[rm]{bf}{sb}` would use `sb` (semi-bold) when `\rmfamily \bfseries` is asked for.

If used without the optional argument, e.g., `\DeclareFontSeriesDefault{bf}{b}` then this is like redefining `\bfdefault` or `\mddefault`.

If some family specify defaults aren't given, e.g. if there are no declarations for, say, `tt` then the format defaults of `\mddefault` and `\bfdefault` are assumed. If those are later changed this is *not* reflected![14]

`\DeclareFontSeriesDefault` The command to declare font series defaults for the "rm", "sf" or "tt" family.

```
\let\DeclareFontSeriesDefault\@undefined      % for rollback
\newcommand\DeclareFontSeriesDefault[3][]{%
  \def\reserved@a{#1}%
```

No optional argument: set up general default.

```
  \ifx\reserved@a\@empty
    \ifcsname #2series\endcsname           % supported are
                                           % \[md/bf]default
```

Adding `\@empty` allows us to detect if the default gets redefined with `\renewcommand` or `\def` by the user.

```
      \expandafter\def
        \csname #2default\endcsname{#3\@empty}%
      \expandafter\def
        \csname #2default@previous\endcsname{#3\@empty}%
    \else

       \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
          {Mandatory first argument must be 'md'  or 'bf'.}
    \fi
```

Optional argument given, set up specific default.

```
  \else
    \ifcsname #2series@#1\endcsname          % supported are
                                             % \[md/bf]series@[rm/sf/tt]
      \expandafter\edef
         \csname #2series@#1\endcsname{#3}%
```

If the interface is used we remove the frozen kernel default. This way, we know that something was explicitly set up (even if the setup has the same value as the default).

```
      \expandafter\let
         \csname #2series@#1@kernel\endcsname\@undefined
    \else
       \@latex@error{Wrong syntax for \string\DeclareFontSeriesDefault}%
          {Optional argument must be 'rm', 'sf', or 'tt'. \MessageBreak
           Mandatory first argument must be 'md'  or 'bf'.}
    \fi
  \fi
}
```

(*End definition for* `\DeclareFontSeriesDefault`*. This function is documented on page* **??**.)

`\mdseries@rm` `\mdseries@sf` `\mdseries@tt` `\bfseries@rm` `\bfseries@sf` `\bfseries@tt` We initialize the family specific default at the end of the format generation. Later on they may get overwritten in the preamble or a package via `\DeclareFontSeriesDefault` (or possibly directly).

Conceptual change: The `\bfdefault` will be `b` not `bx` because that is what it should be really for nearly every font except Computer/Latin Modern.

[14] I see no easy way to achieve this without compromising compatibility with existing packages that currently use mweights and directly define (some) of the `\mdseries@..` commands but not others.

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

This means that it behaves like before because if the default fonts are used then `\bfseries@rm` etc kick in and make `\textbf` use `bx`. However, if the font gets changed then `\bfdefault` will get used.

```
\def\bfseries@rm{bx}
\def\bfseries@sf{bx}
\def\bfseries@tt{bx}
```

Frozen version of the kernel defaults so we can see if they have changed.

```
\let\bfseries@rm@kernel\bfseries@rm
\let\bfseries@sf@kernel\bfseries@sf
\let\bfseries@tt@kernel\bfseries@tt
```

The default for the medium series is `m` and this will be interpreted as resetting both weight and width. To reset only one of them the virtual value `?m` and `m?` are available.

```
\def\mdseries@rm{m}
\def\mdseries@sf{m}
\def\mdseries@tt{m}
```

(*End definition for* `\mdseries@rm` *and others. These functions are documented on page* ??.)

`\series@change@debug` For debugging, but right now none of this code is extracted. The idea is to have a separate package with debugging code one day.

```
⟨*debug⟩
\let\series@change@debug\typeout
\let\series@change@debug\@gobble
⟨/debug⟩
```

(*End definition for* `\series@change@debug`. *This function is documented on page* ??.)

`\prepare@family@series@update` This is core command that prepares for the family update. The big difference to the documented code above is that the nested `\ifx` statements seem to be missing. Instead we loop through an internal list that holds the names of the three meta families. This approach allows us to extend the mechanism at a later stage to allow for additional named meta families.

Here is the current definition of that list:

`\@meta@family@list`

```
\def\@meta@family@list{\@elt{rm}\@elt{sf}\@elt{tt}}
```

```
\def\prepare@family@series@update#1#2{%
 \if@forced@series
⟨+debug⟩ \series@change@debug{No series preparation (forced \f@series)\on@line}%
   \fontfamily#2%
 \else
⟨+debug⟩ \series@change@debug{Prepearing for switching to #1 (#2)\on@line}%
   \expand@font@defaults
```

We prepare for changing the current series. We have to find it before changing the family as discussed above.

```
   \let\target@series@value\@empty
   \def\target@meta@family@value{#1}%
```

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

- `\bfseries` is called for a family using `bx` (e.g., CMR)
- Switch to a font family that is none of the meta families, e.g., via `\fontfamily{ptm}\selectfont`
- Then none of the real meta families, match but the final `\@elt{??}` will.
- Therefore if the current series is `\mddefault` or `\bfdefault` it will be detected and the corresponding target series selected.

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

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

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

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

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

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

```
  \fontfamily#2%
```

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

```
  \ifx\target@series@value\@empty
⟨+debug⟩ \series@change@debug{Target series still empty ...}%
  \else
    \ifx \f@series\target@series@value
⟨+debug⟩ \series@change@debug{Target series unchanged:
⟨+debug⟩                      \f@series \space = \target@series@value}%
    \else
      \maybe@load@fontshape
⟨+debug⟩ \series@change@debug{Target series:
⟨+debug⟩                      \f@series \space -> \target@series@value}%
```

The `\target@series@value` may contain something like `cm` (coming from a default) and so we can't directly asign it to `\f@series` be have to drop any surplus `m` first.

```
%      \let\f@series\target@series@value
       \series@maybe@drop@one@m\target@series@value\f@series
    \fi
  \fi
 \fi
}
```

(*End definition for* `\prepare@family@series@update` *and* `\@meta@family@list`*. These functions are documented on page* ??.)

`\update@series@target@value`

In this macro used in the look you basically find the nested `\ifx`s from the outline above. The only difference is that is it is parameterized instead of being written out and only for one block of tests because the code is called reatedly when looping over the meta family list. From the list we get each meta family name in turn.

```
\def\update@series@target@value#1{%
```

There is one additional test at the beginning, because the list contains all meta families and we need to ignore the case where current one from the list and target one are identical.

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

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

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

If that's the case we know that this is the block that applies (only one meta family can match). So to speed things up we change `\@elt` so that the rest of the loop gets gobbled.

```
      \let\@elt\@gobble
```

Then we try to find the right new value for the series (as explained above). The two macros defined first are only there because we now need to use `\csname` and this way the code will be a little faster.

```
      \expandafter\let\expandafter\reserved@b
                      \csname mdseries@\target@meta@family@value\endcsname
      \expandafter\let\expandafter\reserved@c
                      \csname bfseries@\target@meta@family@value\endcsname
⟨+debug⟩\series@change@debug{Targets for mdseries and bfseries:
⟨+debug⟩                     \reserved@b\space and \reserved@c}%
```

This here is now identical to the nested `\ifx` block from the outline, except that it there appeared twice in `\rmfamily`. This is now covered by looping and stopping the loop when a match was found.

We have to sanitize the default value first because it may contain something like `mc` and that would never match `\f@series` because there it would be called `c` with the `m` dropped. It would be probably better to do that differently these days, but it is hard to adjust without causing a lot of issues, so we do the dropping in various places instead.

```
      \expandafter\series@maybe@drop@one@m
          \csname mdseries@#1\endcsname\reserved@d
      \ifx\reserved@d\f@series
⟨+debug⟩   \series@change@debug{mdseries@#1 matched -> \reserved@b}%
                                      \let\target@series@value\reserved@b
      \else
```

Again do some sanitizing.

```
        \expandafter\series@maybe@drop@one@m
           \csname bfseries@#1\endcsname\reserved@d
        \ifx\reserved@d\f@series
⟨+debug⟩  \series@change@debug{bfseries@#1 matched -> \reserved@c}%
                                      \let\target@series@value\reserved@c
      \else\ifx\f@series\mddef@ult    \let\target@series@value\reserved@b
⟨+debug⟩  \series@change@debug{mddef@ult matched -> \reserved@b}%
      \else\ifx\f@series\bfdef@ult    \let\target@series@value\reserved@c
⟨+debug⟩  \series@change@debug{bfdef@ult matched -> \reserved@c}%
      \fi\fi\fi\fi
    \fi
  \fi
}
```

(*End definition for* `\update@series@target@value`. *This function is documented on page* **??**.)

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

```
\def\init@series@setup{%
```

We only want bx in \bfseries@rm if the roman font is Computer Modern or Latin Modern, otherwise it should be b. It was set to bx in the kernel so that any font use with the default families in the preamble get this value. Now at the real document start we check if the fonts have been changed. If there was a \DeclareFontSeriesDefault declaration or \bfseries@rm was directly altered then it differs from \bfseries@rm@kernel and we do nothing. Otherwise we check if \rmdefault is one of the CM/LM font families and if so we keep bx otherwise we change it to b.

This approach doesn't cover one case: CM/LM got changed to a different family that supports bx, but the support package for that family used \def\bfseries@rm{bx} instead of using \DeclareFontSeriesDefault. In that case the code here changes it to b. Solution: use the \DeclareFontSeriesDefault interface.

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

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

```
  \ifx\bfseries@sf@kernel\bfseries@sf
    \expandafter\in@\expandafter{\sfdefault}%
                     {cmr,cmss,cmtt,lcmss,lcmtt,lmr,lmss,lmtt}%
    \ifin@ \else \def\bfseries@sf{b}\fi\fi
  \ifx\bfseries@tt@kernel\bfseries@tt
    \expandafter\in@\expandafter{\ttdefault}%
                     {cmr,cmss,cmtt,lcmss,lcmtt,lmr,lmss,lmtt}%
    \ifin@ \else \def\bfseries@tt{b}\fi\fi
```

If the document preamble has changed the \familydefault or if the if the \rmdefault contains a new font family, we may have to adjust the series defaults accordingly, before starting typesetting.

Similarly, if the user has changed the \mddefault or the medium series for the family selected as document font we may also have to adjust the \seriesdefault.

On the other hand if the document font is still CM or LM then \bfdefault is wrong, because it is now saying b and not bx as it should for such fonts.

To fix all this we first run \reset@font (the internal kernel name for \normalfont). This will set up the document encoding, family, series and shape based on the current values of \encodingdefault, \familydefault, \seriesdefault and \shapedefault. However, if the family (from \familydefault) has special medium default we should switch to that (and not use what is current value from \seriesdefault). This can be achieved by afterwards calling \mediumseries and then changing \seriesdefault to the now current series value (in \f@series).

But what should happen if \seriesdefault got explicitly changed? In that case the explicit change should surive and we should not alter \seriesdefault. This is solved by comparing the current value of \seriesdefault with a kernel version saved in the format and if they differ we do not call \mdseries or change \seriesdefault.

```
  \reset@font
  \ifx\seriesdefault\seriesdefault@kernel
    \mdseries
    \let\seriesdefault\f@series
  \fi
}%
```

(*End definition for* `\init@series@setup`. *This function is documented on page* **??**.)

As the kernel code now implements the same functionality as mweights, albeit internally coded slightly differently, that package shouldn't be loaded any more. We therefore pretend that it already got loaded. Thus, a font package that tries to load it and then sets `\mdseries@..`, etc. will continue to work but will now use the kernel code.

Of course, mid-term such package should probably use `\DeclareFontSeriesDefault` instead of making using low-level definitions.

```
\expandafter\let\csname ver@mweights.sty\endcsname\fmtversion
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\DeclareFontSeriesDefault}{Custom series}%
⟨latexrelease⟩
⟨latexrelease⟩\let\DeclareFontSeriesDefault\@undefined
⟨latexrelease⟩\let\bfseries@rm\@undefined
⟨latexrelease⟩\let\bfseries@sf\@undefined
⟨latexrelease⟩\let\bfseries@tt\@undefined
⟨latexrelease⟩\let\bfseries@rm@kernel\@undefined
⟨latexrelease⟩\let\bfseries@sf@kernel\@undefined
⟨latexrelease⟩\let\bfseries@tt@kernel\@undefined
⟨latexrelease⟩\let\mdseries@rm\@undefined
⟨latexrelease⟩\let\mdseries@sf\@undefined
⟨latexrelease⟩\let\mdseries@tt\@undefined
⟨latexrelease⟩\expandafter\let\csname ver@mweights.sty\endcsname\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\@meta@family@list\@undefined
⟨latexrelease⟩\let\prepare@family@series@update\@undefined
⟨latexrelease⟩\let\update@series@target@value\@undefined
⟨latexrelease⟩
```

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

```
⟨latexrelease⟩\let\init@series@setup\relax
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\bfseries}{Custom series with hooks}%
```

`\expand@font@defaults` `\rm@def@ult` `\sf@def@ult` `\tt@def@ult` `\md@def@ult` `\bf@def@ult`

The family specific defaults are fully expanded, i.e., they are defined via `\edef` inside `\DeclareFontSeriesDefault`. However, the overall defaults, e.g., `\bfdefault` may have been redefined by the user and thus may not be fully expanded. So to enable reliable comparison we make expanded versions of them. That we rerun each time. The alternative would be to only allow for changes before begin document.

```
\def\expand@font@defaults{%
  \edef\rmdef@ult{\rmdefault}%
  \edef\sfdef@ult{\sfdefault}%
  \edef\ttdef@ult{\ttdefault}%
```

The series defaults may contain some surplus `m` that we need to drop here.

```
  \series@maybe@drop@one@m\bfdefault\bfdef@ult
  \series@maybe@drop@one@m\mddefault\mddef@ult
```

Formats that set up parallel fonts, e.g., for Japanese, can use this hook to add additional code here.

```
  \UseHook{expand@font@defaults}%
}
```

(*End definition for* `\expand@font@defaults` *and others. These functions are documented on page* **??**.)

`\bfseries` This document command switches to the bold series.

```
\DeclareRobustCommand\bfseries{%
  \not@math@alphabet\bfseries\mathbf
```

In the original NFSS definition it then called `\fontseries` with the value `\bfdefault`. In the new scheme we have more alternatives and therefore check if the current family (`\f@family`) is the current `\rmdef@ult`, `\sfdef@ult` or `\ttdef@ult` and the select the correct family default in that case.

```
  \expand@font@defaults
```

If `\bfdefault` and `\bfdefault@previous` are different then the default got changed directly through the legacy interface (i.e., via `\def` or `\renewcommand`. In that case we reset all meta family defaults so that the document behaves like it was the case before the new mechanism was introduced.

```
  \ifx\bfdefault\bfdefault@previous\else
```

We add `\@empty` and then let `\bfdefault@previous` to `\bfdefault` so that we can detect any further change.

```
    \expandafter\def\expandafter\bfdefault
                    \expandafter{\bfdefault\@empty}%
    \let\bfdefault@previous\bfdefault
```

And we reset the meta family defaults (`\bfdef@ult` is an expanded version of `\bfdefault`.

```
    \let\bfseries@rm\bfdef@ult
    \let\bfseries@sf\bfdef@ult
    \let\bfseries@tt\bfdef@ult
```

Formats that set up parallel fonts, e.g., for Japanese, can use this hook to add resets here. Not that this hook is only run when resets are necessary.

```
    \UseHook{bfseries/defaults}%
  \fi
```

```
    \ifx\f@family\rmdef@ult      \fontseries\bfseries@rm
    \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
    \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt
```

If not `\bfdefault` is used.

```
    \else                        \fontseries\bfdefault
    \fi\fi\fi
```

This hook in contrast is always executed.

```
  \UseHook{bfseries}%
  \selectfont
}
```

(*End definition for* `\bfseries`. *This function is documented on page* **??**.)

\mdseries This document command switches to the medium series.

```
\DeclareRobustCommand\mdseries{%
  \not@math@alphabet\mdseries\relax
  \expand@font@defaults
  \ifx\mddefault\mddefault@previous\else
    \expandafter\def\expandafter\mddefault\expandafter{\mddefault\@empty}%
    \let\mddefault@previous\mddefault
    \let\mdseries@rm\mddef@ult
    \let\mdseries@sf\mddef@ult
    \let\mdseries@tt\mddef@ult
```

Formats that set up parallel fonts, e.g., for Japanese, can use this hook to add resets here.

```
    \UseHook{mdseries/defaults}%
  \fi
    \ifx\f@family\rmdef@ult      \fontseries\mdseries@rm
    \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
    \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
    \else                        \fontseries\mddefault
    \fi\fi\fi
  \UseHook{mdseries}%
  \selectfont
}
```

(*End definition for* `\mdseries`*. This function is documented on page* ??.)

\rmfamily Here are the document level commands for changing the main font families, or rather, here is a documented outline of the code, the actual code is then streamlined and somewhat generalized.

```
DeclareRobustCommand\rmfamily{%
 \not@math@alphabet\rmfamily\mathrm
```

If families are changed then we have to do a bit more work. In the original NFSS implementation a family change kept encoding, series shape and size unchanged but now we can't any longer simply reuse the current series value. Instead we may have to change it from one family default to the next.

```
 \expand@font@defaults
```

We have to do the testing while the current family is still unchanged but we have to do the adjustment of the series after it got changed (because the new family might has different sets ofshapes available and we certainly don't want to see substituation going on. So we use `\target@series@value` to hold the target series (if any).

```
 \let\target@series@value\@empty
```

Thus, if the current family is the sans family

```
 \ifx\f@family\sfdef@ult
```

and if we using the medium series of the sans family

```
    \ifx\f@series\mdseries@sf
```

then lets switch to the medium series for the serif family

```
                                        \let\target@series@value\mdseries@rm
```

and if we use the bold series of the sans family switch to the bold default of the serif family:

```
      \else\ifx\f@series\bfseries@sf  \let\target@series@value\bfseries@rm
```

However, the sans family may not have any specific defaults set, so we also compare with the overall defaults.

```
      \else\ifx\f@series\mddef@ult    \let\target@series@value\mdseries@rm
      \else\ifx\f@series\bfdef@ult    \let\target@series@value\bfseries@rm
```

If neither test was true we leave the series alone. This way a special manual setting such as `\fontseries{lc}` is not undone if the family changes (of course there may not be any support for it in the new family but then the NFSS substitution kicks in and sorts it out).

```
      \fi\fi\fi\fi
```

We need to do the same if the current family is the typewriter family:

```
 \else\ifx\f@family\ttdef@ult
      \ifx\f@series\mdseries@tt       \let\target@series@value\mdseries@rm
      \else\ifx\f@series\bfseries@tt  \let\target@series@value\bfseries@rm
      \else\ifx\f@series\mddef@ult    \let\target@series@value\mdseries@rm
      \else\ifx\f@series\bfdef@ult    \let\target@series@value\bfseries@rm
      \fi\fi\fi\fi
 \fi\fi
```

With these preparations for series out of the way we can now change the font family to `\rmdefault`.

```
 \fontfamily\rmdefault
```

If `\target@series@value` is still empty there is nothing more to do other than selecting the new family. However, if not then we should update the font series now as well. But there is one further subtle issue. We may not have loaded an `.fd` file for our target font family yet. In the past that was done in `\selectfont` if necessary but since we are now doing all the comparisons in `\fontseries` we need to make sure that the font family specifications are already loaded prior to calling `\fontseries`.

```
 \ifx\target@series@value\@empty \else
   \maybe@load@fontshape
```

Updating the series in this case means directly changing `\f@series` to the target value. We don't want to go through `\fontseries` because that would apply the mappings and then `bx + b` would keep `bx` instead of changing to `b` as desired. as

```
   \let\f@series\target@series@value
 \fi
 \selectfont}
```

So now for the real definition: most of the code above gets delegated to a helper command `\prepare@family@series@update` so that the definition becomes again fairly short. In addition we add a hook, mainly for our Japanese friends so that the code can be extended prior to the call to `\selectfont`.

```
\DeclareRobustCommand\rmfamily{%
   \not@math@alphabet\rmfamily\mathrm
```

This holds all the code discussed above, first argument is the meta family, i.e., `rm` in this case, and second argument is the default family name, e.g., `cmr` indirectly accessed via `\rmdefault`. This is calling `\fontfamily` and if necessary `\fontseries` as outline above.

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

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

```
   \UseHook{rmfamily}%
   \selectfont}
```

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

`\sffamily`
`\ttfamily`

```
\DeclareRobustCommand\sffamily{%
   \not@math@alphabet\sffamily\mathsf
   \prepare@family@series@update{sf}\sfdefault
   \UseHook{sffamily}%
   \selectfont}

\DeclareRobustCommand\ttfamily{%
   \not@math@alphabet\ttfamily\mathtt
   \prepare@family@series@update{tt}\ttdefault
   \UseHook{ttfamily}%
   \selectfont}
```

(*End definition for* `\rmfamily`*,* `\sffamily`*, and* `\ttfamily`*. These functions are documented on page* **??**.)

`rmfamily`
`sffamily`
`ttfamily`
`normalfont`
`expand@font@defaults`
`bfseries`
`bfseries/defaults`
`mdseries`
`mdseries/defaults`

Declare the hooks used above.

```
\NewHook{rmfamily}
\NewHook{sffamily}
\NewHook{ttfamily}
\NewHook{normalfont}
\NewHook{expand@font@defaults}
\NewHook{bfseries}
\NewHook{bfseries/defaults}
\NewHook{mdseries}
\NewHook{mdseries/defaults}
```

(*End definition for* `rmfamily` *and others. These functions are documented on page* **??**.)

`\@rmfamilyhook`
`\@sffamilyhook`
`\@ttfamilyhook`
`\@defaultfamilyhook`

These four hooks have legacy versions used in 2020/02/02 so we should support them until they aren't any longer used.

By default the hooks do nothing.

```
\let\@rmfamilyhook\@empty
\let\@sffamilyhook\@empty
\let\@ttfamilyhook\@empty
\let\@defaultfamilyhook\@empty   %FMi sort out
```

(*End definition for* `\@rmfamilyhook` *and others. These functions are documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                 {\bfseries}{Custom series with hooks}%
⟨latexrelease⟩
⟨latexrelease⟩\def\expand@font@defaults{%
⟨latexrelease⟩  \edef\rmdef@ult{\rmdefault}%
⟨latexrelease⟩  \edef\sfdef@ult{\sfdefault}%
⟨latexrelease⟩  \edef\ttdef@ult{\ttdefault}%
⟨latexrelease⟩  \edef\bfdef@ult{\bfdefault}%
⟨latexrelease⟩  \edef\mddef@ult{\mddefault}%
⟨latexrelease⟩  \edef\famdef@ult{\familydefault}%
⟨latexrelease⟩}
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\bfseries{%
⟨latexrelease⟩  \not@math@alphabet\bfseries\mathbf
⟨latexrelease⟩  \expand@font@defaults
⟨latexrelease⟩    \ifx\f@family\rmdef@ult      \fontseries\bfseries@rm
⟨latexrelease⟩    \else\ifx\f@family\sfdef@ult \fontseries\bfseries@sf
⟨latexrelease⟩    \else\ifx\f@family\ttdef@ult \fontseries\bfseries@tt
⟨latexrelease⟩    \else                        \fontseries\bfdefault
⟨latexrelease⟩    \fi\fi\fi
⟨latexrelease⟩  \selectfont
⟨latexrelease⟩}
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\mdseries{%
⟨latexrelease⟩  \not@math@alphabet\mdseries\relax
⟨latexrelease⟩  \expand@font@defaults
⟨latexrelease⟩    \ifx\f@family\rmdef@ult      \fontseries\mdseries@rm
⟨latexrelease⟩    \else\ifx\f@family\sfdef@ult \fontseries\mdseries@sf
⟨latexrelease⟩    \else\ifx\f@family\ttdef@ult \fontseries\mdseries@tt
⟨latexrelease⟩    \else                        \fontseries\mddefault
⟨latexrelease⟩    \fi\fi\fi
⟨latexrelease⟩  \selectfont
⟨latexrelease⟩}
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\rmfamily{%
⟨latexrelease⟩   \not@math@alphabet\rmfamily\mathrm
⟨latexrelease⟩   \prepare@family@series@update{rm}\rmdefault
⟨latexrelease⟩   \@rmfamilyhook
⟨latexrelease⟩   \selectfont}
⟨latexrelease⟩\DeclareRobustCommand\sffamily{%
⟨latexrelease⟩   \not@math@alphabet\sffamily\mathsf
⟨latexrelease⟩   \prepare@family@series@update{sf}\sfdefault
⟨latexrelease⟩   \@sffamilyhook
⟨latexrelease⟩   \selectfont}
⟨latexrelease⟩\DeclareRobustCommand\ttfamily{%
⟨latexrelease⟩   \not@math@alphabet\ttfamily\mathtt
⟨latexrelease⟩   \prepare@family@series@update{tt}\ttdefault
⟨latexrelease⟩   \@ttfamilyhook
⟨latexrelease⟩   \selectfont}
⟨latexrelease⟩
```

```
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\bfseries}{Custom series with hooks}%
⟨latexrelease⟩
⟨latexrelease⟩\let\expand@font@defaults\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\bfseries
⟨latexrelease⟩        {\not@math@alphabet\bfseries\mathbf
⟨latexrelease⟩         \fontseries\bfdefault\selectfont}
⟨latexrelease⟩\DeclareRobustCommand\mdseries
⟨latexrelease⟩        {\not@math@alphabet\mdseries\relax
⟨latexrelease⟩         \fontseries\mddefault\selectfont}
⟨latexrelease⟩\DeclareRobustCommand\rmfamily
⟨latexrelease⟩        {\not@math@alphabet\rmfamily\mathrm
⟨latexrelease⟩         \fontfamily\rmdefault\selectfont}
⟨latexrelease⟩\DeclareRobustCommand\sffamily
⟨latexrelease⟩        {\not@math@alphabet\sffamily\mathsf
⟨latexrelease⟩         \fontfamily\sfdefault\selectfont}
⟨latexrelease⟩\DeclareRobustCommand\ttfamily
⟨latexrelease⟩        {\not@math@alphabet\ttfamily\mathtt
⟨latexrelease⟩         \fontfamily\ttdefault\selectfont}
⟨latexrelease⟩
⟨latexrelease⟩\let\@rmfamilyhook\@undefined
⟨latexrelease⟩\let\@sffamilyhook\@undefined
⟨latexrelease⟩\let\@ttfamilyhook\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\IfFontSeriesContextTF` With the ability for `\bfseries` or `\mdseries` to be mapped to different NFSS axis values it becomes important to have the ability to determine the current context as we can no longer look at `\f@series` to answer a question such as "am I currently typsetting in a bold typeface?"

This is provided by the test `\IfFontSeriesContextTF`. It takes three arguments:

- The context we try to check (either `bf` for bold or `md` for medium, i.e., the same that can go into the first mandatory argument of `\DeclareFontSeriesDefault`),
- what to do if we are in this context (true case) and
- what to do if we are not (false case).

This allows you to define commands like `\IfBold`, e.g.,

```
\newcommand\IfBold[2]{\IfSeriesContextTF{bf}{#1}{#2}}
```

and then do

```
This is \IfBold{bold}{non-bold} text.
```

and get the appropriate result.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\IfFontSeriesContextTF}{Font series context}%
```

```
\DeclareRobustCommand\IfFontSeriesContextTF[1]{%
  \expand@font@defaults
```

In the beginning we haven't found the context we are loking for.

```
  \@font@series@contextfalse
```

We store the requested context away for use in the tests.

```
  \def\requested@test@context{#1}%
```

The next definition is there to ensure that get a final match during testing even if the current family is non of the meta families (`rm`, `sf` or `tt`). This will then basically tests if the current font family matches the overall default.

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

Then we run through the meta family list (currently containing just the three values) followed by the artifical meta family `??` and test each of them in turn using `\test@font@series@context` as the testing command.

```
  \let\@elt\test@font@series@context
      \@meta@family@list
      \@elt{??}%
  \let\@elt\relax
```

Following that we evaluate the status of `\if@font@series@context` to determine which of the remaining arguments (true/false case) we have to execute.

```
  \if@font@series@context
  \expandafter\@firstoftwo
  \else
  \expandafter\@secondoftwo
  \fi
}
```

(*End definition for* `\IfFontSeriesContextTF`*. This function is documented on page* **??**.)

`\test@font@series@context` This tests the context (stored in `\requested@test@context`) and updates the bookean if the right context is found.

```
\def\test@font@series@context#1{%
```

First task is to figure out whether the current family matches `\rmfamily`, `\sffamily`, etc. so in `\reserved@a` we store the value of `\rmdef@ult` (or whatever the given meta family is) and compare that to `\f@family`.

```
  \edef\reserved@a{\csname #1def@ult\endcsname}%
  \ifx\f@family\reserved@a
```

If they match we have found the right meta family so we don't need to test any of the remaining meta family and therefore change `\@elt` to `\@gobble`.

```
    \let\@elt\@gobble
```

Now we have to test if `\f@series` matches the requested context (e.g., whether `\bfseries@rm` has that value if the current meta family is `rm` and we are looking for the `bf` context).

```
    \expandafter\ifx
                \csname\requested@test@context series@#1\endcsname\f@series
```

If yes we change the boolean and are done.

```
      \@font@series@contexttrue
```

If not then maybe the reason is that nothing special was set up for that meta family so we also check now check if `\f@series` matches the overall default (e.g., `\bfdef@ult` if we are looking for the bold context). If that matches we change the boolean.

```
    \else
      \expandafter\ifx
                  \csname\requested@test@context def@ult\endcsname\f@series
        \@font@series@contexttrue
  \fi\fi\fi
}
```

(*End definition for* `\test@font@series@context`*. This function is documented on page* **??**.)

`\if@font@series@context` The boolean to signal if we found the requested font series context.

```
\newif\if@font@series@context
```

(*End definition for* `\if@font@series@context`*. This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\IfFontSeriesContextTF}{Font series context}%
⟨latexrelease⟩
⟨latexrelease⟩\let\IfFontSeriesContextTF\@undefined
⟨latexrelease⟩\let\test@font@series@context\@undefined
⟨latexrelease⟩\let\if@font@series@context\@undefined
⟨latexrelease⟩\let\@font@series@contexttrue\@undefined
⟨latexrelease⟩\let\@font@series@contextfalse\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

3 Supporting nested emphasis

By default LaTeX 2_ε supports two levels of nested emphasis: if the current font has an upright shape then it switches to `\itshape` otherwise to `\eminnershape` (which defaults to `\upshape`). This means nested emphasis will ocssilate between italic and upright shapes.

Sometimes it would be nice to allow for a more lengthly sequence, but instead of providing a fixed one LaTeX now offers a general mechanism that allows to define arbitrary sequences.

`\DeclareEmphSequence`
`\emforce`

This declaration expects a comma separated list of (font) change declarations corresponding to increasing levels of emphasis. The mechanism tries to be "smart" and verifies that the declarations actually alter the font. If not it will ignore this level and tries the next one—the assumption being that there was a manual font change in the document to the font that is now supposed to be used for emphasis. Of course, this only works if the declarations in the list actually change the font and not, say, just the color. In such a case one has to use `\emforce` to which directs the mechanism to use the level even if the font attributes haven't changed.

`\emreset` If the nesting is so deep, that the specified levels are exhausted then `\emreset` is used as a final set of declarations (which by default returns back to the upright shape). Any additional nesting levels will then reuse the list from its beginning.

\DeclareEmphSequence

\DeclareEmphSequence expects a clist of declaration. Spaces in the argument are dropped to avoid surious spaces in the output. The declarations are additive. At the very end the shape is reset using \emreset and \emforce so that this case is never skipped.[15] Further nested calls restart at the beginning.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                    {\DeclareEmphSequence}{Nested emph}%
\def\DeclareEmphSequence#1{%
  \protected@edef\emfontdeclare@clist{\zap@space#1, \@empty\emforce\emreset}%
}
```

By default the it is empty, in which case \eminnershape is used by LaTeX.

```
\let\emfontdeclare@clist\@empty
```

(*End definition for* \DeclareEmphSequence*. This function is documented on page* ??.)

\emrest

Reset the font to upright and upper/lower case. With the default rules using \shapedefault does that for us but to be on the safe side we do it like this:

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

(*End definition for* \emrest*. This function is documented on page* ??.)

\em

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

```
\DeclareRobustCommand\em{%
  \@nomath\em
  \ifx\emfontdeclare@clist\@empty
    \ifdim \fontdimen\@ne\font >\z@
      \eminnershape \else \itshape \fi
  \else
```

But if not we use the list to decide how to do emphasis.

We use the current font to check if the declarations have any effect, so even a size change is allowed and identified as a modification (but a color change, for example, isn't). So first we save the current status.

```
  \edef\em@currfont{\csname\curr@fontshape/\f@size\endcsname}%
```

Then we grab the next element from the list and check if it can be used.

```
    \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
  \fi
}
```

```
\def\eminnershape{\upshape}
```

(*End definition for* \em*. This function is documented on page* ??.)

\do@emfont@update

We know that the list (if not empty) has at least 2 elements separated by a comma, so we pick up the first in #1 and the rest in #2.

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

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

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

[15] Maybe we should not add \emforce but allow that case to be skipped as well. Of course, that might result in an endless loop if somebody defines a sequence without any font change and without \emforce but ...

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

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

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

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

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

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

```
                            \expandafter\endcsname
                    \em@currfont
  \expandafter\do@emfont@update\emfontdeclare@clist\do@emfont@update
```

If `\emforce` was used, we have to undo its effect:

```
  \else
    \let\em@force\@empty
  \fi
}
```

(*End definition for* `\do@emfont@update`*. This function is documented on page* **??**.)

`\emforce`
`\em@force`

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

```
\protected\def\emforce{\def\em@force{/}}
\let\em@force\@empty
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\emforce` *and* `\em@force`*. These functions are documented on page* **??**.)

`\em`
`\eminnershape`

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

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`.

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\DeclareEmphSequence}{Nested emph}%
⟨latexrelease⟩\let\DeclareEmphSequence\@undefined
⟨latexrelease⟩\let\emfontdeclare@clist\@undefined
⟨latexrelease⟩\let\emreset\@undefined
⟨latexrelease⟩\let\do@emfont@update\@undefined
⟨latexrelease⟩\let\emforce\@undefined
⟨latexrelease⟩\let\em@force\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\em
⟨latexrelease⟩        {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
⟨latexrelease⟩                        \eminnershape \else \itshape \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\DeclareEmphSequence}{Nested emph}%
```

```
⟨latexrelease⟩\DeclareRobustCommand\em
⟨latexrelease⟩        {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
⟨latexrelease⟩                        \upshape \else \itshape \fi}%
⟨latexrelease⟩\let\eminnershape\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\em` *and* `\eminnershape`*. These functions are documented on page* **??**.)

`\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}
```

(*End definition for* `\not@math@alphabet`*. This function is documented on page* **??**.)

Finally we provide two abbreviations to switch to the LaTeX *versions*.

```
\DeclareRobustCommand\boldmath{\@nomath\boldmath
              \mathversion{bold}}
\DeclareRobustCommand\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}
```

3.1 Legacy

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.

`\newfont`

```
\def\newfont#1#2{\@ifdefinable#1{\font#1=#2\relax}}
```

(*End definition for* `\newfont`*. This function is documented on page* **??**.)

\symbol

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\symbol}{XeTeX change for math}%
\ifdefined\XeTeXversion
  \DeclareRobustCommand\symbol[1]{\Ucharcat#1 12\relax}
\else
  \DeclareRobustCommand\symbol[1]{\char#1\relax}
\fi
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\symbol}{XeTeX change for math}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\symbol[1]{\char#1\relax}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \symbol*. This function is documented on page* ??.)

3.2 Miscellaneous

\@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⟩
```

(*End definition for* \@setfontsize *and* \@setsize*. These functions are documented on page* ??.)

\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}
```

(*End definition for* \hexnumber@*. This function is documented on page* ??.)

\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}}}
```

(*End definition for* \nfss@text. *This function is documented on page* ??.)

\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}}}
```

(*End definition for* \copyright. *This function is documented on page* ??.)

\normalfont \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:

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\normalfont}{Add hook to \normalfont}%
\DeclareRobustCommand\normalfont{%
```

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

```
    \fontencoding\encodingdefault
    \edef\f@family{\familydefault}%
    \edef\f@series{\seriesdefault}%
    \edef\f@shape{\shapedefault}%

    \UseHook{normalfont}%
```

This is the old name for the hook introduced in 2020/02/02. It will be removed in one of the future releases!

```
    \@defaultfamilyhook        % hookname from 2020/02 will vanish
    \selectfont}
\let\reset@font\normalfont
```

(*End definition for* \normalfont *and* \reset@font. *These functions are documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\normalfont}{Add hook to \normalfont}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\normalfont
⟨latexrelease⟩          {\usefont\encodingdefault
⟨latexrelease⟩                   \familydefault
```

```
⟨latexrelease⟩                    \seriesdefault
⟨latexrelease⟩                    \shapedefault
⟨latexrelease⟩                    \relax}
⟨latexrelease⟩\let\reset@font\normalfont
⟨latexrelease⟩
⟨latexrelease⟩\let\@defaultfamilyhook\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

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 also set some default values for `\f@family` etc. Note that we don't yet have any encodings that comes later. In the past this was implicitly done by `\DeclareErrorFont`.

```
\fontfamily{cmr}
\fontseries{m}
\fontshape{n}
\fontsize{10}{10}
```

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

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

We now load the customizable parts of NFSS.

```
\InputIfFileExists{fonttext.cfg}
           {\typeout{=====================================^^J%
                     ^^J%
```

```
                      Local config file fonttext.cfg used^^J%
                     ^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{fonttext.ltx}}
\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%
                     ^^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%
                     ^^J%
                     ====================================}%
             \def\@addtofilelist##1{\xdef\@filelist{\@filelist,##1}}%
            }
           {\input{preload.ltx}}
\let\@addtofilelist\@gobble
```

`\seriesdefault` `\seriesdefault@kernel`

After `\seriesdefault` got defined inside `fonttext.ltx` or a `.cfg` file overwriting it, we alter its value by appending `\@empty` to it. This will vanish if expanded but allows us to check if the default gets altered (even to the same value) in the document preamble. All we have to do is to save the current value somewhere and later compare the two. For this we use `\seriesdefault@kernel`.

```
\expandafter\def\expandafter\seriesdefault\expandafter{\seriesdefault\@empty}
\let\seriesdefault@kernel\seriesdefault
```

(*End definition for* `\seriesdefault` *and* `\seriesdefault@kernel`*. These functions are documented on page* **??**.)

`\@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\=
```

(*End definition for* `\@acci`*,* `\@accii`*, and* `\@acciii`*. These functions are documented on page* **??**.)

`\cal` `\mit`

Here were the two old ⟨*alphabet identifiers*⟩.

(*End definition for* `\cal` *and* `\mit`*. These functions are documented on page* **??**.)

```
⟨/2ekernel⟩
```

File x
fontdef.dtx

<-latexrelease> [2020/08/01 v3.0i LaTeX Kernel (<-latexrelease> font setup)]

1 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.

2 Customization

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

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

Notice that some of these encodings are 'old' in the sense that we hope that they will be superseded soon by encoding standards defined by the 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.

3 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}}
```

4 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⟩
```

5 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}
```

5.1 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 {omsenc.def}
```

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

```
\input {ot1enc.def}
\input  {t1enc.def}
\input{ts1enc.def}
```

```
\ifx\Umathcode\@undefined
```

We then set the default text font encoding. This will hopefully change some day to `T1`. This setting should *not* be changed to produce a portable format.

```
\fontencoding{OT1}
```

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

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

```
\def\rmsubstdefault{cmr}
\def\sfsubstdefault{cmss}
\def\ttsubstdefault{cmtt}
\LoadFontDefinitionFile{TS1}{cmr}
```

```
\else
```

Unicode.

```
\input {tuenc.def}
\fontencoding{TU}
```

The initial `fontenc` package load list if a Unicode engine is used:

```
\def\@fontenc@load@list{\@elt{TU}}
```

```
\DeclareFontSubstitution{TU}{lmr}{m}{n}
\LoadFontDefinitionFile{TU}{lmr}
\LoadFontDefinitionFile{TU}{lmss}
\LoadFontDefinitionFile{TU}{lmtt}
```

```
\def\rmsubstdefault{lmr}
\def\sfsubstdefault{lmss}
\def\ttsubstdefault{lmtt}
\LoadFontDefinitionFile{TS1}{lmr}
```

```
\DeclareFontSubstitution{TU}{lmr}{m}{n}
```

End of Unicode branch.

```
\fi
```

If different encodings for text fonts are in use one could put the common setup into `\DeclareFontEncodingDefaults`. There is now a better mechanism so using this interface is discouraged!

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

Then we define the default substitution for every encoding. This release of LaTeX 2_ε assumes that the ec fonts are available. It is possible to change this to point to some other font family (e.g., Times with the appropriate encoding if it is available) without making documents non-portable. However, in such a case documents will produce different page breaks at other sites. The substitution defaults can all be changed without losing portability as long as there are font shape definitions for the selected substitutions.

```
\DeclareFontSubstitution{T1}{cmr}{m}{n}
\DeclareFontSubstitution{OT1}{cmr}{m}{n}
```

For every encoding declaration, LaTeX 2_ε will try to verify that the given substitution information makes sense, i.e. that it is impossible to go into an endless loop if font substitution happens. This is done at the moment the `\begin{document}` is encountered. LaTeX 2_ε will then check that for every encoding the substitution defaults form a valid font shape group, which means that it will check if there is a `\DeclareFontShape` declaration for this combination. We will therefore load the corresponding `.fd` files now. If we don't do this they would be loaded at verification time (i.e. at `\begin{document}` which would delay processing unnecessarily.

> **Warning:** Please note that this means that you have to regenerate the format whenever you change any of these `.fd` files since LaTeX 2_ε will not read `.fd` files if it already knows about the encoding/family combination.

The `\nfss@catcodes` ensures that white space is ignored in any definitions made in the fd files.

```
\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.

5.2 Defaults

To allow the use of `\rmfamily`, `\sffamily`, etc. in documents even if non-standard families are used we provide nine macros which hold the name of the corresponding families, series, and so on. This makes it easy to use other font families (like Times Roman, etc.). One simply has to redefine these defaults.

All these hooks have to be defined in this file but you can change their meaning (except for `\encodingdefault`) without making documents non-portable.

`\encodingdefault` `\rmdefault` `\sfdefault` `\ttdefault`

The following three definitions set up the meaning for `\rmfamily`, `\sffamily`, and `\ttfamily`.

```
\ifx\Umathcode\@undefined
\newcommand\encodingdefault{OT1}
\newcommand\rmdefault{cmr}
\newcommand\sfdefault{cmss}
\newcommand\ttdefault{cmtt}
\else
\newcommand\encodingdefault{TU}
\newcommand\rmdefault{lmr}
\fontfamily{\rmdefault}
\newcommand\sfdefault{lmss}
\newcommand\ttdefault{lmtt}
\fi
⟨/text⟩
⟨latexrelease⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease⟩                  {\encodingdefault}{TU encoding default}%
⟨latexrelease⟩\ifx\Umathcode\@undefined
⟨latexrelease⟩\renewcommand\encodingdefault{OT1}
⟨latexrelease⟩\fontencoding{\encodingdefault}
⟨latexrelease⟩\renewcommand\rmdefault{cmr}
⟨latexrelease⟩\fontfamily{\rmdefault}
⟨latexrelease⟩\renewcommand\sfdefault{cmss}
⟨latexrelease⟩\renewcommand\ttdefault{cmtt}
⟨latexrelease⟩\else
⟨latexrelease⟩\renewcommand\encodingdefault{TU}
⟨latexrelease⟩%done in everyjob\fontencoding{\encodingdefault}
⟨latexrelease⟩\renewcommand\rmdefault{lmr}
⟨latexrelease⟩\fontfamily{\rmdefault}
⟨latexrelease⟩\renewcommand\sfdefault{lmss}
⟨latexrelease⟩\renewcommand\ttdefault{lmtt}
⟨latexrelease⟩\fi
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\encodingdefault}{TU encoding default}%
```

```
⟨latexrelease⟩\fontencoding{OT1}
⟨latexrelease⟩\renewcommand\encodingdefault{OT1}
⟨latexrelease⟩\fontencoding{\encodingdefault}
⟨latexrelease⟩\renewcommand\rmdefault{cmr}
⟨latexrelease⟩\fontfamily{\rmdefault}
⟨latexrelease⟩\renewcommand\sfdefault{cmss}
⟨latexrelease⟩\renewcommand\ttdefault{cmtt}
⟨latexrelease⟩\EndIncludeInRelease
⟨*text⟩
```

(*End definition for* `\encodingdefault` *and others. These functions are documented on page* **??**.)

`\bfdefault` `\mddefault`

Series changing commands are influenced by the following hooks.

```
\newcommand\bfdefault{b}  % overwritten below (for rollback)
\newcommand\mddefault{m}  % overwritten below (for rollback)
```

(*End definition for* `\bfdefault` *and* `\mddefault`*. These functions are documented on page* **??**.)

`\itdefault` `\sldefault` `\scdefault` `\updefault`

Shape changing commands use the following hooks.

```
\newcommand\itdefault{it}
\newcommand\sldefault{sl}
\newcommand\scdefault{sc}
\newcommand\updefault{up}  % overwritten below (for rollback)
```

(*End definition for* `\itdefault` *and others. These functions are documented on page* **??**.)

```
⟨/text⟩
⟨*text | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\updefault}{font defaults change}%
%     \begin{macrocode}
\renewcommand\updefault{up}
```

We append `\@empty` to the series value so that we can detect if it got changed via `\def` or `\renewcommand` later.

```
\renewcommand\bfdefault{b\@empty}
\renewcommand\mddefault{m\@empty}
```

```
\let\bfdefault@previous\bfdefault
\let\mddefault@previous\mddefault
⟨/text | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\updefault}{font defaults change}%
⟨latexrelease⟩
⟨latexrelease⟩\renewcommand\updefault{n}
⟨latexrelease⟩\renewcommand\bfdefault{bx}
⟨latexrelease⟩
⟨latexrelease⟩\let\bfdefault@previous\undefined
⟨latexrelease⟩\let\mddefault@previous\undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*text⟩
```

\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\familydefault{\rmdefault}
\newcommand\seriesdefault{\mddefault}
```

In previous releases `\shapedefault` pointed to `\updefault` which resolved to `n`, but these days that is no longer the case (and `up` is wrong when you want to do a reset. So we now use `n` explicitly.

```
\newcommand\shapedefault{n}
```

(*End definition for* `\familydefault`, `\seriesdefault`, *and* `\shapedefault`. *These functions are documented on page* ??.)

This finishes the low-level setup in `fonttext.ltx`.

```
⟨/text⟩
```

6 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}
```

6.1 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
```

6.1.1 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}
```

6.2 Math font sizes

The declarations below declare the text, script and scriptscript size to be used for each text font size.

All occurrences of sizes longer than a single character are replaced with the macro name that holds them, saving a number of tokens (but losing a bit of speed, so this may not stay this way).

```
\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}
```

6.3 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.

6.3.1 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}
```

6.3.2 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}
```

6.3.3 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 % \_
```

6.3.4 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!

6.4 Symbols accessed via control sequences

6.4.1 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}
```

6.4.2 Ordinary symbols

```
\DeclareMathSymbol{\aleph}{\mathord}{symbols}{"40}
\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}
\DeclareMathSymbol{\top}{\mathord}{symbols}{"3E}
\DeclareMathSymbol{\bot}{\mathord}{symbols}{"3F}
\DeclareMathSymbol{\triangle}{\mathord}{symbols}{"34}
\DeclareMathSymbol{\forall}{\mathord}{symbols}{"38}
\DeclareMathSymbol{\exists}{\mathord}{symbols}{"39}
\DeclareMathSymbol{\neg}{\mathord}{symbols}{"3A}
```

Alias:

```
%    \let\lnot=\neg
\DeclareMathSymbol{\lnot}{\mathord}{symbols}{"3A}
```

```
\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}
```

```
\DeclareRobustCommand\hbar{{\mathchar'26\mkern-9muh}}
\DeclareRobustCommand\surd{{\mathchar"1270}}
\DeclareRobustCommand\angle{{\vbox{\ialign{$\m@th\scriptstyle##$\crcr
      \not\mathrel{\mkern14mu}\crcr
      \noalign{\nointerlineskip}
      \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crcr}}}}
```

6.4.3 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}
    \DeclareRobustCommand\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}
    \DeclareRobustCommand\oint{\ointop\nolimits}
\DeclareMathSymbol{\bigsqcup}{\mathop}{largesymbols}{"46}
\DeclareMathSymbol{\smallint}{\mathop}{symbols}{"73}
```

6.4.4 Binary symbols

```
\DeclareMathSymbol{\triangleleft}{\mathbin}{letters}{"2F}
\DeclareMathSymbol{\triangleright}{\mathbin}{letters}{"2E}
\DeclareMathSymbol{\bigtriangleup}{\mathbin}{symbols}{"34}
\DeclareMathSymbol{\bigtriangledown}{\mathbin}{symbols}{"35}
```

Alias:

```
%   \let \varbigtriangledown \bigtriangledown
%   \let \varbigtriangleup \bigtriangleup
\DeclareMathSymbol{\varbigtriangleup}{\mathbin}{symbols}{"34}
\DeclareMathSymbol{\varbigtriangledown}{\mathbin}{symbols}{"35}
```

These last two synonyms are needed because the stmaryrd package redefines them as Operators.

```
\DeclareMathSymbol{\wedge}{\mathbin}{symbols}{"5E}
\DeclareMathSymbol{\vee}{\mathbin}{symbols}{"5F}
```

Alias:

```
%   \let\land=\wedge
%   \let\lor=\vee
\DeclareMathSymbol{\land}{\mathbin}{symbols}{"5E}
\DeclareMathSymbol{\lor}{\mathbin}{symbols}{"5F}
```

```
\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}
```

6.4.5 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}
   \DeclareRobustCommand\neq{\not=}
```

As `\neq` is robust we should not use `\let` to define `\ne` as then it would change if `\neq` changes.

```
   \DeclareRobustCommand\ne{\not=}
```

It would ok to use `\let` for those declared by `\DeclareMathSymbol` but for a cleaner interface we avoid it always (just in case the internals change).

```
\DeclareMathSymbol{\leq}{\mathrel}{symbols}{"14}
\DeclareMathSymbol{\geq}{\mathrel}{symbols}{"15}
```

Alias:

```
%   \let\le=\leq
%   \let\ge=\geq
\DeclareMathSymbol{\le}{\mathrel}{symbols}{"14}
\DeclareMathSymbol{\ge}{\mathrel}{symbols}{"15}
```

```
\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}
```

Alias:

```
%    \let\owns=\ni
```

```
\DeclareMathSymbol{\owns}{\mathrel}{symbols}{"33}
\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}
\DeclareMathSymbol{\rightarrow}{\mathrel}{symbols}{"21}
```

Alias:

```
%   \let\gets=\leftarrow
%   \let\to=\rightarrow
\DeclareMathSymbol{\gets}{\mathrel}{symbols}{"20}
\DeclareMathSymbol{\to}{\mathrel}{symbols}{"21}
\DeclareMathSymbol{\mapstochar}{\mathrel}{symbols}{"37}
   \DeclareRobustCommand\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=}
```

6.4.6 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}
   \DeclareRobustCommand\hookrightarrow{\lhook\joinrel\rightarrow}
\DeclareMathSymbol{\rhook}{\mathrel}{letters}{"2D}
   \DeclareRobustCommand\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\;}
```

6.4.7 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}}
```

6.4.8 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}
```

6.4.9 Radicals

```
\DeclareMathRadical{\sqrtsign}{symbols}{"70}{largesymbols}{"70}
```

6.4.10 Over and under something, etc

```
\DeclareRobustCommand\overrightarrow[1]{\vbox{\m@th\ialign{##\crcr
      \rightarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\DeclareRobustCommand\overleftarrow[1]{\vbox{\m@th\ialign{##\crcr
      \leftarrowfill\crcr\noalign{\kern-\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}
\DeclareRobustCommand\overbrace[1]
     {\mathop{\vbox{\m@th\ialign{##\crcr\noalign{\kern3\p@}%
      \downbracefill\crcr\noalign{\kern3\p@\nointerlineskip}%
      $\hfil\displaystyle{#1}\hfil$\crcr}}}\limits}
\DeclareRobustCommand\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)

```
\DeclareRobustCommand\skew[3]
  {{\muskip\z@#1mu\divide\muskip\z@\tw@ \mkern\muskip\z@
    #2{\mkern-\muskip\z@{#3}\mkern\muskip\z@}\mkern-\muskip\z@}{}}
\DeclareRobustCommand\rightarrowfill{$\m@th\smash-\mkern-7mu%
  \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
  \mkern-7mu\mathord\rightarrow$}
\DeclareRobustCommand\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}
\DeclareRobustCommand\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$}
\DeclareRobustCommand\upbracefill{$\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$}
```

6.4.11 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}
```

`\DeclareMathDelimiter` produces a command that is robust (with an internal macro containing the payload) so we should not use `\let` for making an alias

```
%\let\|=\Vert
\DeclareMathDelimiter{\|}
   {\mathord}{symbols}{"6B}{largesymbols}{"0D}
\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}
```

(*End definition for* \lgroup*,* \rgroup*, and* \bracevert*. These functions are documented on page* ??.)

6.5 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}
\DeclareRobustCommand\mathsterling{\mathit{\mathchar"7024}}
\DeclareRobustCommand\mathunderscore{\kern.06em\vbox{\hrule\@width.3em}}
```

(*End definition for* \mathparagraph *and others. These functions are documented on page* ??.)

\mathellipsis

This is plain TEX's \ldots.

```
\DeclareRobustCommand\mathellipsis{\mathinner{\ldotp\ldotp\ldotp}}%
```

(*End definition for* \mathellipsis*. This function is documented on page* ??.)

6.6 Other special functions and parameters

6.6.1 Biggggg

```
⟨/math⟩
⟨*math | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                    {\Big}{Start LR-mode}%
\DeclareRobustCommand\big[1]{\leavevmode@ifvmode
   {\hbox{$\left#1\vbox to8.5\p@{}\right.\n@space$}}}
\DeclareRobustCommand\Big[1]{\leavevmode@ifvmode
   {\hbox{$\left#1\vbox to11.5\p@{}\right.\n@space$}}}
\DeclareRobustCommand\bigg[1]{\leavevmode@ifvmode
   {\hbox{$\left#1\vbox to14.5\p@{}\right.\n@space$}}}
\DeclareRobustCommand\Bigg[1]{\leavevmode@ifvmode
   {\hbox{$\left#1\vbox to17.5\p@{}\right.\n@space$}}}
⟨/math | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\Big}{Start LR-mode}%
⟨latexrelease⟩\def\big#1{{\hbox{$\left#1\vbox to8.5\p@{}\right.\n@space$}}}
⟨latexrelease⟩\def\Big#1{{\hbox{$\left#1\vbox to11.5\p@{}\right.\n@space$}}}
⟨latexrelease⟩\def\bigg#1{{\hbox{$\left#1\vbox to14.5\p@{}\right.\n@space$}}}
⟨latexrelease⟩\def\Bigg#1{{\hbox{$\left#1\vbox to17.5\p@{}\right.\n@space$}}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*math⟩
```

```
\def\n@space{\nulldelimiterspace\z@ \m@th}
```

6.6.2 The log-like functions

`\operator@font` The `\operator@font` determines the symbol font used for log-like functions.

```
\def\operator@font{\mathgroup\symoperators}
```

(*End definition for* `\operator@font`*. This function is documented on page* **??**.)

6.6.3 Parameters

```
\thinmuskip=3mu
\medmuskip=4mu plus 2mu minus 4mu
\thickmuskip=5mu plus 5mu
```

This finishes the low-level setup in `fontmath.ltx`.

```
⟨/math⟩
```

7 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 y
preload.dtx

1 Overview

This file contains an number of possible settings for preloading fonts during installation of NFSS2 (which is used by LaTeX 2_ε). It will be used to generate the following files:

preload.min	minimal subset of fonts necessary to run NFSS2
preload.ori	preload of CM fonts similar to the old `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.

2 Customization

You can customize the preloaded fonts in your LaTeX 2_ε system by installing a file with the name `preload.cfg`. If this file exists it will be used in place of the system file `preload.ltx`. You can, for example, copy one of the files mentioned above (that can be generated from this source) to `preload.cfg`.

Or you can define completely other preloads. In that case start from `preload.min` since that contains the fonts that have to be preloaded by *all* LaTeX 2_ε systems.

Avoid using `preload.ori`, it will load so many fonts that on most installations it is nearly impossible to load other font families afterwards. This file is only generated to show what fonts have been preloaded by 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.

3 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.

4 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⟩
```

5 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⟩
```

File z
ltfntcmd.dtx

Abstract

The commands defined in this file `ltfntcmd` are part of the kernel code for LaTeX 2_ε/NFSS2.
It is also meant to serve as documentation for package writers since it demonstrates how to define high-level font changing commands using a small number of creator functions.

1 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`.[16] 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
```

[16] 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.

Command	*Corresponds to*	*Action*
`\textnormal{..}`	`\normalfont`	Typeset argument in normal family
`\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.

```
\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 `\/`.

2 The implementation

`\DeclareTextFontCommand` This is the creator function for `\text..` commands. It gives a warning if `\foo` or `\fragfoo` is already defined.

In math mode it simply puts the font declaration and text into a box (possibly an automagically sized one).

Otherwise it first scans the text to see where `\nocorr` occurs within it. This sets the `\check@ic` commands to do what is necessary concerning the italic correction at both ends.

The algorithm for deciding whether to put in an italic correction is not very subtle: one is added whenever the newly current font is not itself positively sloped, unless the next token is a character in the 'nocorr' list. At the end of the text this is done after closing the group so as to check the 'outer font'. Note that this will often result in adding an italic correction token after a character in an unsloped font; we believe (in early 2003) that this is perhaps inefficient but not dangerous.

It also now checks for empty contents of the text command and optimises this case. Some care is also taken to check that doing dangerous things in vertical mode is avoided.

The italic correction token is added to the horizontal list before (in the list) an immediately preceding non-zero glob of glue (skip) and any non-zero penalty preceding that since, in the typical case, this puts it immediately after the last character in the preceding word.

Note that it is necessary to put in the `\aftergroup\maybe@ic` at the end of the group so that it comes after any other aftergroup tokens and immediately before the following tokens. It is also necessary to remove the `\fi` from the token list before the group ends; this is done by adding an `\expandafter` just before the closing brace.

```
⟨*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
                        }%
}
```

(*End definition for* `\DeclareTextFontCommand`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\textrm` *and others. These functions are documented on page* **??**.)

`\textbf` `\textmd`

For the series attribute:

```
\DeclareTextFontCommand{\textbf}{\bfseries}
\DeclareTextFontCommand{\textmd}{\mdseries}
```

(*End definition for* `\textbf` *and* `\textmd`*. These functions are documented on page* **??**.)

`\textit` `\textsl` `\textsc` `\textup`

And for the shapes:

```
\DeclareTextFontCommand{\textit}{\itshape}
\DeclareTextFontCommand{\textsl}{\slshape}
\DeclareTextFontCommand{\textsc}{\scshape}
\DeclareTextFontCommand{\textup}{\upshape}
```

(*End definition for* `\textit` *and others. These functions are documented on page* **??**.)

`textulc` `textsw` `textssc`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\textulc}{Additional text commands}%
\DeclareTextFontCommand{\textulc}{\ulcshape}
\DeclareTextFontCommand{\textsw}{\swshape}
\DeclareTextFontCommand{\textssc}{\sscshape}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\textulc}{Additional text commands}%
⟨latexrelease⟩
⟨latexrelease⟩\let\textulc\@undefined
⟨latexrelease⟩\let\textsw\@undefined
⟨latexrelease⟩\let\textssc\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `textulc`*,* `textsw`*, and* `textssc`*. These functions are documented on page* **??**.)

`\emph`

Finally we have the `\em` font change declaration of LaTeX. The corresponding definition with argument is

```
\DeclareTextFontCommand{\emph}{\em}
```

(*End definition for* `\emph`*. This function is documented on page* **??**.)

`\nocorr`

This is just a label, so it does nothing; it should also be unexpandable.

```
\let \nocorr \relax
```

(*End definition for* `\nocorr`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\check@icl` *and* `\check@icr`*. These functions are documented on page* **??**.)

`\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
}
```

(*End definition for* \text@command *and* \check@nocorr@*. These functions are documented on page* ??.)

\ifmaybe@ic Switch used soley within \maybe@ic not interfering with other switches.

```
\newif\ifmaybe@ic
```

(*End definition for* \ifmaybe@ic*. This function is documented on page* ??.)

\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
}
```

(*End definition for* \maybe@ic *and* \maybe@ic@*. These functions are documented on page* ??.)

\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
}
```

(*End definition for* \t@st@ic*. This function is documented on page* ??.)

\sw@slant The definition of the mysterious \sw@slant command is as follows.

\fix@penalty

```
\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
}
```

(*End definition for* `\sw@slant` *and* `\fix@penalty`*. These functions are documented on page* **??**.)

`\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 {,.}
```

(*End definition for* `\nocorrlist`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\nfss@text`*. This function is documented on page* **??**.)

`\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}}%
}
```

(*End definition for* `\DeclareOldFontCommand`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* \@fontswitch*,* \@@math@egroup*, and* \@@math@egroup*. These functions are documented on page* ??.)

These commands are available only in the preamble.

```
\@onlypreamble \DeclareTextFontCommand
\@onlypreamble \DeclareOldFontCommand
```

3 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⟩
```

(*End definition for* \normalsize*. This function is documented on page* ??.)

File A
lttextcomp.dtx

This file contains the implementation for accessing the glyphs provided by the TS1 encoding (Text Companion Encoding). This is now offered as part of the kernel and so the textcomp package which used to provide the definitions is now mainly needed for compatibility reasons (and doesn't do much any more).

\oldstylenums
\legacyoldstylenums

Preserve the old definition of \oldstylenums under a different name.

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.

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\oldstylenums}{Old style numerals}%
\DeclareRobustCommand\legacyoldstylenums[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
}
```

And here is the improved one that adjusts depending on surroundings.

```
\DeclareRobustCommand\oldstylenums[1]{%
 \begingroup
 \ifmmode
   \mathgroup\symletters #1%
  \else
```

The \CheckEncodingSubset is discused below.

```
   \CheckEncodingSubset\@use@text@encoding{TS1}\tc@oldstylesubst2{{#1}}%
 \fi
 \endgroup
}
```

The helper to select the substitution if needed.

```
\def\tc@oldstylesubst#1{%
   \tc@errorwarn
          {Oldstyle digits unavailable for
           family \f@family.\MessageBreak
           Default oldstyle digits used instead}\@eha
  \bgroup
      \expand@font@defaults
```

The substitution defaults are provided in the file `fonttext.ltx`.

```
      \ifx\f@family\rmdef@ult
         \fontfamily\rmsubstdefault
      \else\ifx\f@family\sfdef@ult
         \fontfamily\sfsubstdefault
      \else\ifx\f@family\ttdef@ult
         \fontfamily\ttsubstdefault
      \else
        \fontfamily\textcompsubstdefault
      \fi\fi\fi
      \fontencoding{TS1}\selectfont#1%
  \egroup
}
```

(*End definition for* `\oldstylenums` *and* `\legacyoldstylenums`*. These functions are documented on page* ??.)

`\textcompsubstdefault` Here is the default for the "unknown" case:

```
\def\textcompsubstdefault{\rmsubstdefault}
```

(*End definition for* `\textcompsubstdefault`*. This function is documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\oldstylenums}{Old style numerals}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\oldstylenums[1]{%
⟨latexrelease⟩    \begingroup
⟨latexrelease⟩     \spaceskip\fontdimen\tw@\font
⟨latexrelease⟩     \usefont{OML}{\rmdefault}{\f@series}{it}%
⟨latexrelease⟩     \mathgroup\symletters #1%
⟨latexrelease⟩    \endgroup
⟨latexrelease⟩}
⟨latexrelease⟩\let\legacyoldstylenums\@undefined
⟨latexrelease⟩\def\textcompsubstdefault{cmr}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

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

```
⟨*2ekernel⟩
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩    {\DeclareEncodingSubset}{Text companion symbols}%
```

`\DeclareEncodingSubset` The declaration takes 3 mandatory arguments: an *encoding* for which a subsetting is wanted (currently always `TS1`, and most likely forever), the *font family* for which we declare the subset and finally the *subset* number (between `0` (all of the encoding is supported) and `9` many glyphs are missing.

For `TS1` the numbers have been choosen in a way that most fonts can be fairly correctly categorized, but the default settings are always conservative, that is they may claim that less glyphs are supported than there actually are.

As these days many font families are set up to end in `-LF` (lining figures), `-OsF` (oldstyle figures), etc. the declaration supports a shortcut: if the *font family* name ends in `-*` then the star gets replaced by these common ending, e.g.,

```
\DeclareEncodingSubset{TS1}{Alegreya-*}{2}
```

is the same as writing

```
\DeclareEncodingSubset{TS1}{Alegreya-LF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-OsF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-TLF}{2}
\DeclareEncodingSubset{TS1}{Alegreya-TOsF}{2}
```

If only some are needed then one can define them individually but in many cases all four are wanted, hence the shortcut.

The coding of the declaration has no error checking as it is mostly for internal use.

```
\def\DeclareEncodingSubset#1#2{%
  \DeclareEncodingSubset@aux{#1}#2*\DeclareEncodingSubset@aux
}
```

```
\def\DeclareEncodingSubset@aux#1#2*#3\DeclareEncodingSubset@aux#4{%
```

if `#3` is empty then there was no star, otherwise we define all four variants.

```
  \expandafter\ifx\expandafter X\detokenize{#3}X%
    \@DeclareEncodingSubset{#1}{#2}{#4}%
  \else
    \@DeclareEncodingSubset{#1}{#2LF}{#4}%
    \@DeclareEncodingSubset{#1}{#2TLF}{#4}%
    \@DeclareEncodingSubset{#1}{#2OsF}{#4}%
    \@DeclareEncodingSubset{#1}{#2TOsF}{#4}%
  \fi
}
```

The subset info is stored in a command with the name `\`*family*`:`*subset* so if that already exists we change otherwise declare a subset.

```
\def\@DeclareEncodingSubset#1#2#3{%
   \@ifundefined{#1:#2}%
     {\@font@info{Setting #2 sub-encoding to #1/#3}}%
     {\@font@info{Changing #2 sub-encoding to #1/#3}}%
     \@namedef{#1:#2}{#3}}
```

Any reason to allow those in the middle of documents?

```
\@onlypreamble\DeclareEncodingSubset
\@onlypreamble\DeclareEncodingSubset@aux
\@onlypreamble\@DeclareEncodingSubset
```

(*End definition for* `\DeclareEncodingSubset`*. This function is documented on page* **??**.)

`\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.

```
\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%
}
```

(*End definition for* `\CheckEncodingSubset`*. This function is documented on page* **??**.)

To set up the glyphs for the subsets we need a number helpers.

`\tc@errorwarn` To we produce errors, warnings, or only info in the transcripts if glyphs require substitutions? By default it is "info" only. With the `textcomp` package that can be changed.

```
\def\tc@errorwarn#1#2{\@latex@info{#1}}
```

(*End definition for* `\tc@errorwarn`*. This function is documented on page* **??**.)

`\tc@subst`

```
\def\tc@subst#1{%
   \tc@errorwarn
    {Symbol \string#1 not provided by\MessageBreak
     font family \f@family\space
     in TS1 encoding.\MessageBreak Default family used instead}\@eha
  \bgroup
      \expand@font@defaults
      \ifx\f@family\rmdef@ult
         \fontfamily\rmsubstdefault
      \else\ifx\f@family\sfdef@ult
         \fontfamily\sfsubstdefault
      \else\ifx\f@family\ttdef@ult
         \fontfamily\ttsubstdefault
      \else
         \fontfamily\textcompsubstdefault
      \fi\fi\fi
```

Whatever default was chosen, we claim now (locally hopefully) that it can handle all slots (even if not true) to avoid looping in certain situations, e.g., when something was set up incorrectly.

```
      \@namedef{TS1:\f@family}{0}%
      \selectfont#1%
  \egroup
}
```

(*End definition for* `\tc@subst`. *This function is documented on page* ??.)

`\tc@fake@euro` `\tc@fake@euro` is an example of a "fake" definition to use in arg `#3` of `\CheckEncodingSubset` when a symbol is not available in a certain font family. Here we produce a poor man's Euro symbol by combining a "C" with a "=".

```
\def\tc@fake@euro#1{%
   \leavevmode
   \@font@info{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
   }%
}
```

(*End definition for* `\tc@fake@euro`. *This function is documented on page* ??.)

`\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}
```

Accents and been mad an error in the `textcomp` package when not available. Now that we provide the functionality in the kernel we avoid the eror by swapping in a `T1` accent if the `TS1` accent is not available.

```
%\def\tc@check@accent{\CheckEncodingSubset\UseTextAccent{TS1}\tc@error}
\def\tc@check@accent#1{\CheckEncodingSubset\UseTextAccent{TS1}{\tc@swap@accent#1}}
\def\tc@swap@accent#1#2{\UseTextAccent{T1}#1}
```

(*End definition for* `\tc@check@symbol` *and* `\tc@check@accent`. *These functions are documented on page* ??.)

1 Sub-encodings

Here are the default definitions for the `TS1` symbols. First those that we assume are always available if a font implements `TS1`.

```
\DeclareTextSymbolDefault{\textdollar}{TS1}
\UndeclareTextCommand{\textdollar}  {OT1}         % don't use the OT1 def any longer
\DeclareTextSymbolDefault{\textsterling}{TS1}
\UndeclareTextCommand{\textsterling}{OT1}         % don't use the OT1 def any longer
\DeclareTextSymbolDefault{\textperthousand}{TS1}
\UndeclareTextCommand{\textperthousand}{T1}       % don't use the T1 def
```

Using \UndeclareTextCommand above is enough only if the encoding definition files are not reloaded afterwards. In the past that happened if fontenc was used in the document preamble (not any longer). So in some sense it is better to fully remove them from the encoding files, but for rollbacks it is easier to keep them in for now.

These are the standard itemize and footnote symbols originally taken from OMS and now from TS1:

```
\DeclareTextSymbolDefault{\textasteriskcentered}{TS1}
\DeclareTextSymbolDefault{\textbullet}{TS1}
\DeclareTextSymbolDefault{\textdaggerdbl}{TS1}
\DeclareTextSymbolDefault{\textdagger}{TS1}
\DeclareTextSymbolDefault{\textparagraph}{TS1}
\DeclareTextSymbolDefault{\textperiodcentered}{TS1}
\DeclareTextSymbolDefault{\textsection}{TS1}
```

And here are the other TS1 glyphs that are implemented by every font (or nearly everyone—a few are commented out and moved to sub-encoding 9, because they aren't around in one or two fonts.

```
%%\DeclareTextSymbolDefault{\textbardbl}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\textbrokenbar}{TS1}
%%\DeclareTextSymbolDefault{\textcelsius}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\textcent}{TS1}
\DeclareTextSymbolDefault{\textcopyright}{TS1}
\DeclareTextSymbolDefault{\textdegree}{TS1}
\DeclareTextSymbolDefault{\textdiv}{TS1}
\DeclareTextSymbolDefault{\textlnot}{TS1}
\DeclareTextSymbolDefault{\textonehalf}{TS1}
\DeclareTextSymbolDefault{\textonequarter}{TS1}
%%\DeclareTextSymbolDefault{\textonesuperior}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\textordfeminine}{TS1}
\DeclareTextSymbolDefault{\textordmasculine}{TS1}
\DeclareTextSymbolDefault{\textpm}{TS1}
\DeclareTextSymbolDefault{\textquotesingle}{TS1}
\DeclareTextSymbolDefault{\textquotestraightbase}{TS1}
\DeclareTextSymbolDefault{\textquotestraightdblbase}{TS1}
\DeclareTextSymbolDefault{\textregistered}{TS1}
%%\DeclareTextSymbolDefault{\textthreequartersemdash}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\textthreequarters}{TS1}
%%\DeclareTextSymbolDefault{\textthreesuperior}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\texttimes}{TS1}
\DeclareTextSymbolDefault{\texttrademark}{TS1}
%%\DeclareTextSymbolDefault{\texttwelveudash}{TS1} % subst in sub-enc 9 above
%%\DeclareTextSymbolDefault{\texttwosuperior}{TS1} % subst in sub-enc 9 above
\DeclareTextSymbolDefault{\textyen}{TS1}
```

```
\DeclareTextSymbolDefault{\textcapitalcompwordmark}{TS1}
\DeclareTextSymbolDefault{\textascendercompwordmark}{TS1}
```

In the following sections the remaining default definitions are ordered by sub-encoding in which they are become unavailable (i.e., they are not provided in the sub-encoding with that number and all sub-encodings with higher numbers.

Thus the symbols that are available in sub-encoding x are the symbols above (always available) and the symbols list in the sections for sub-encodings $x + 1$ and higher.

1.1 Sub-encoding 1 (drop symbols not working in Latin Modern)

The \textcircled is available but the glyph is simply too small so we keep using the OMS glyph.

```
\DeclareTextCommandDefault{\textcircled}
  {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OMS}}1\textcircled}
```

1.2 Sub-encoding 2 (majority of new OTF fonts via autoinst)

```
\DeclareTextCommandDefault{\t}
  {\CheckEncodingSubset\UseTextAccent{TS1}{\UseTextAccent{OML}}2\t}
```

Capital accents are really only very seldom implemented, so from sub-encoding 2 onwards we use the normal T1 accents if they are asked for in the document.

In Unicode engines we don't implement them at all but always use the basic accents instead. whether that works or not really depends on the font, something like \"X usually comes out wrong in Unicode engines.

```
\ifx\Umathcode\@undefined
  \DeclareTextCommandDefault{\capitalacute}       {\tc@check@accent{\'}2\capitalacute}
  \DeclareTextCommandDefault{\capitalbreve}       {\tc@check@accent{\u}2\capitalbreve}
  \DeclareTextCommandDefault{\capitalcaron}       {\tc@check@accent{\v}2\capitalcaron}
  \DeclareTextCommandDefault{\capitalcedilla}     {\tc@check@accent{\c}2\capitalcedilla}
  \DeclareTextCommandDefault{\capitalcircumflex}  {\tc@check@accent{\^}2\capitalcircumflex}
  \DeclareTextCommandDefault{\capitaldieresis}    {\tc@check@accent{\"}2\capitaldieresis}
  \DeclareTextCommandDefault{\capitaldotaccent}   {\tc@check@accent{\.}2\capitaldotaccent}
  \DeclareTextCommandDefault{\capitalgrave}       {\tc@check@accent{\`}2\capitalgrave}
  \DeclareTextCommandDefault{\capitalhungarumlaut}{\tc@check@accent{\H}2\capitalhungarumlaut}
  \DeclareTextCommandDefault{\capitalmacron}      {\tc@check@accent{\=}2\capitalmacron}
  \DeclareTextCommandDefault{\capitalogonek}      {\tc@check@accent{\k}2\capitalogonek}
  \DeclareTextCommandDefault{\capitalring}        {\tc@check@accent{\r}2\capitalring}
  \DeclareTextCommandDefault{\capitaltie}         {\tc@check@accent{\t}2\capitaltie}
  \DeclareTextCommandDefault{\capitaltilde}       {\tc@check@accent{\~}2\capitaltilde}
```

For \newtie and \capitalnewtie this is actually wrong, they should pick up the accent from the substitution font (not done yet).

```
  \DeclareTextCommandDefault{\newtie}             {\tc@check@accent{\t}2\newtie}
  \DeclareTextCommandDefault{\capitalnewtie}      {\tc@check@accent{\t}2\capitalnewtie}
```

In Unicode engines we just execute the simple accents:

```
\else
  \DeclareTextCommandDefault\capitalacute{\@tabacckludge'}
  \DeclareTextCommandDefault\capitalbreve{\u}
  \DeclareTextCommandDefault\capitalcaron{\v}
  \DeclareTextCommandDefault\capitalcedilla{\c}
  \DeclareTextCommandDefault\capitalcircumflex{\^}
  \DeclareTextCommandDefault\capitaldieresis{\"}
  \DeclareTextCommandDefault\capitaldotaccent{\.}
  \DeclareTextCommandDefault\capitalgrave{\@tabacckludge`}
  \DeclareTextCommandDefault\capitalhungarumlaut{\H}
  \DeclareTextCommandDefault\capitalmacron{\@tabacckludge=}
  \DeclareTextCommandDefault\capitalnewtie{\t}
  \DeclareTextCommandDefault\capitalogonek{\k}
  \DeclareTextCommandDefault\capitalring{\r}
  \DeclareTextCommandDefault\capitaltie{\t}
  \DeclareTextCommandDefault\capitaltilde{\~}
```

```
  \DeclareTextCommandDefault\newtie{\t}
\fi
```

The next two symbols exist in some fonts (faked?), but we ignore that to keep the subsets reasonable compact and most important linear.

```
\DeclareTextCommandDefault{\textlbrackdbl}      {\tc@check@symbol2\textlbrackdbl}
\DeclareTextCommandDefault{\textrbrackdbl}      {\tc@check@symbol2\textrbrackdbl}
```

Old style numerals are again in some fonts but using `-OsF`, etc. is the better approach to get them, so we claim they aren't in sub-encoding 2 as that's true for most fonts.

```
\DeclareTextCommandDefault{\texteightoldstyle}  {\tc@check@symbol2\texteightoldstyle}
\DeclareTextCommandDefault{\textfiveoldstyle}   {\tc@check@symbol2\textfiveoldstyle}
\DeclareTextCommandDefault{\textfouroldstyle}   {\tc@check@symbol2\textfouroldstyle}
\DeclareTextCommandDefault{\textnineoldstyle}   {\tc@check@symbol2\textnineoldstyle}
\DeclareTextCommandDefault{\textoneoldstyle}    {\tc@check@symbol2\textoneoldstyle}
\DeclareTextCommandDefault{\textsevenoldstyle}  {\tc@check@symbol2\textsevenoldstyle}
\DeclareTextCommandDefault{\textsixoldstyle}    {\tc@check@symbol2\textsixoldstyle}
\DeclareTextCommandDefault{\textthreeoldstyle}  {\tc@check@symbol2\textthreeoldstyle}
\DeclareTextCommandDefault{\texttwooldstyle}    {\tc@check@symbol2\texttwooldstyle}
\DeclareTextCommandDefault{\textzerooldstyle}   {\tc@check@symbol2\textzerooldstyle}
```

The next set of glyphs is special to TeX fonts (and available with a few older PS fonts supported in the virtual fonts), but not any longer in the majority of fonts provided through autoinst, so we pretend there aren't available in sub-encoding 2 and below.

```
\DeclareTextCommandDefault{\textacutedbl}       {\tc@check@symbol2\textacutedbl}
\DeclareTextCommandDefault{\textasciiacute}     {\tc@check@symbol2\textasciiacute}
\DeclareTextCommandDefault{\textasciibreve}     {\tc@check@symbol2\textasciibreve}
\DeclareTextCommandDefault{\textasciicaron}     {\tc@check@symbol2\textasciicaron}
\DeclareTextCommandDefault{\textasciidieresis}  {\tc@check@symbol2\textasciidieresis}
\DeclareTextCommandDefault{\textasciigrave}     {\tc@check@symbol2\textasciigrave}
\DeclareTextCommandDefault{\textasciimacron}    {\tc@check@symbol2\textasciimacron}
\DeclareTextCommandDefault{\textgravedbl}       {\tc@check@symbol2\textgravedbl}
\DeclareTextCommandDefault{\texttildelow}       {\tc@check@symbol2\texttildelow}
```

Finally those below are only available in CM-based fonts but in no font that has its origin outside of the TEX world.

```
\DeclareTextCommandDefault{\textbaht}           {\tc@check@symbol2\textbaht}
\DeclareTextCommandDefault{\textbigcircle}      {\tc@check@symbol2\textbigcircle}
\DeclareTextCommandDefault{\textborn}           {\tc@check@symbol2\textborn}
\DeclareTextCommandDefault{\textcentoldstyle}   {\tc@check@symbol2\textcentoldstyle}
\DeclareTextCommandDefault{\textcircledP}       {\tc@check@symbol2\textcircledP}
\DeclareTextCommandDefault{\textcopyleft}       {\tc@check@symbol2\textcopyleft}
\DeclareTextCommandDefault{\textdblhyphenchar}  {\tc@check@symbol2\textdblhyphenchar}
\DeclareTextCommandDefault{\textdblhyphen}      {\tc@check@symbol2\textdblhyphen}
\DeclareTextCommandDefault{\textdied}           {\tc@check@symbol2\textdied}
\DeclareTextCommandDefault{\textdiscount}       {\tc@check@symbol2\textdiscount}
\DeclareTextCommandDefault{\textdivorced}       {\tc@check@symbol2\textdivorced}
\DeclareTextCommandDefault{\textdollaroldstyle} {\tc@check@symbol2\textdollaroldstyle}
\DeclareTextCommandDefault{\textguarani}        {\tc@check@symbol2\textguarani}
\DeclareTextCommandDefault{\textleaf}           {\tc@check@symbol2\textleaf}
\DeclareTextCommandDefault{\textlquill}         {\tc@check@symbol2\textlquill}
\DeclareTextCommandDefault{\textmarried}        {\tc@check@symbol2\textmarried}
\DeclareTextCommandDefault{\textmho}            {\tc@check@symbol2\textmho}
\DeclareTextCommandDefault{\textmusicalnote}    {\tc@check@symbol2\textmusicalnote}
\DeclareTextCommandDefault{\textnaira}          {\tc@check@symbol2\textnaira}
\DeclareTextCommandDefault{\textopenbullet}     {\tc@check@symbol2\textopenbullet}
```

```
\DeclareTextCommandDefault{\textpeso}           {\tc@check@symbol2\textpeso}
\DeclareTextCommandDefault{\textpilcrow}        {\tc@check@symbol2\textpilcrow}
\DeclareTextCommandDefault{\textrecipe}         {\tc@check@symbol2\textrecipe}
\DeclareTextCommandDefault{\textreferencemark}  {\tc@check@symbol2\textreferencemark}
\DeclareTextCommandDefault{\textrquill}         {\tc@check@symbol2\textrquill}
\DeclareTextCommandDefault{\textservicemark}    {\tc@check@symbol2\textservicemark}
\DeclareTextCommandDefault{\textsurd}           {\tc@check@symbol2\textsurd}
```

The `\textpertenthousand` also belongs in this group but here we have a choice: in T1 there is definition for `\textpertenthousand` making the symbol up from % and `\char 24` (twice) but in many fonts that char doesn't exist and the slot is reused for random ligatures. So better not use it because often it is wrong. But pointing to TS1 is also not great as only a few fonts have it as a real symbol, so we get a substitution to CM or LM.

Alternatively we could just state that the symbol is unavailable in those fonts. For now I substitute.

```
\DeclareTextCommandDefault{\textpertenthousand} {\tc@check@symbol2\textpertenthousand}
\UndeclareTextCommand{\textpertenthousand}{T1}
```

1.3 Sub-encoding 3

Sub-encoding 2 is the one where we loose many symbols. In the higher-numbered sub-encodings we see only a few dropped additionally.

```
\DeclareTextCommandDefault{\textlangle}         {\tc@check@symbol3\textlangle}
\DeclareTextCommandDefault{\textrangle}         {\tc@check@symbol3\textrangle}
```

1.4 Sub-encoding 4

```
\DeclareTextCommandDefault{\textcolonmonetary}  {\tc@check@symbol4\textcolonmonetary}
\DeclareTextCommandDefault{\textdong}           {\tc@check@symbol4\textdong}
\DeclareTextCommandDefault{\textdownarrow}      {\tc@check@symbol4\textdownarrow}
\DeclareTextCommandDefault{\textleftarrow}      {\tc@check@symbol4\textleftarrow}
\DeclareTextCommandDefault{\textlira}           {\tc@check@symbol4\textlira}
\DeclareTextCommandDefault{\textrightarrow}     {\tc@check@symbol4\textrightarrow}
\DeclareTextCommandDefault{\textuparrow}        {\tc@check@symbol4\textuparrow}
\DeclareTextCommandDefault{\textwon}            {\tc@check@symbol4\textwon}
```

1.5 Sub-encoding 5 (most older PS fonts)

Most older PS fonts (supported in TEX since the early nineties when virtual fonts became available) are sorted under this sub-encoding. But in reality, many of them don't have all glpyhs that should be available in sub-encoding 5. Instead they show little squares, i.e., they produce "tofu" if you are unlucky.

But the coverage is so random that it is impossible to sort them properly and if we tried to ensure that they only typeset those glyphs that are really always available wouput put them all into sub-encoding 9 so that's a compromise really.

Modern fonts that don't typeset a tofu character if a glyph is missing are only cataloged as sub-encoding 5 if they really support of its glyph set.

```
\DeclareTextCommandDefault{\textestimated}      {\tc@check@symbol5\textestimated}
\DeclareTextCommandDefault{\textnumero}         {\tc@check@symbol5\textnumero}
```

1.6 Sub-encoding 6

```
\DeclareTextCommandDefault{\textflorin}         {\tc@check@symbol6\textflorin}
\DeclareTextCommandDefault{\textcurrency}       {\tc@check@symbol6\textcurrency}
```

1.7 Sub-encoding 7

```
\DeclareTextCommandDefault{\textfractionsolidus}{\tc@check@symbol7\textfractionsolidus}
\DeclareTextCommandDefault{\textohm}            {\tc@check@symbol7\textohm}
\DeclareTextCommandDefault{\textmu}             {\tc@check@symbol7\textmu}
\DeclareTextCommandDefault{\textminus}          {\tc@check@symbol7\textminus}
```

1.8 Sub-encoding 8

```
\DeclareTextCommandDefault{\textblank}          {\tc@check@symbol{8}\textblank}
\DeclareTextCommandDefault{\textinterrobangdown}{\tc@check@symbol{8}\textinterrobangdown}
\DeclareTextCommandDefault{\textinterrobang}    {\tc@check@symbol{8}\textinterrobang}
```

Fonts with this sub-encoding don't have a Euro symbol, but instead of substituting we fake it.

```
\DeclareTextCommandDefault{\texteuro}
            {\CheckEncodingSubset\UseTextSymbol{TS1}\tc@fake@euro{8}\texteuro}
```

1.9 Sub-encoding 9 (most missing)

```
\DeclareTextCommandDefault{\textcelsius}{\tc@check@symbol{9}\textcelsius}
\DeclareTextCommandDefault{\textonesuperior}{\tc@check@symbol{9}\textonesuperior}
\DeclareTextCommandDefault{\textthreequartersemdash}{\tc@check@symbol{9}\textthreequartersemd
\DeclareTextCommandDefault{\textthreesuperior}{\tc@check@symbol{9}\textthreesuperior}
\DeclareTextCommandDefault{\texttwelveudash}{\tc@check@symbol{9}\texttwelveudash}
\DeclareTextCommandDefault{\texttwosuperior}{\tc@check@symbol{9}\texttwosuperior}
\DeclareTextCommandDefault{\textbardbl}{\tc@check@symbol{9}\textbardbl}
```

2 Unicode engine specials

If we are using a unicode engine we handle some glyphs differently, so this here are the definitions for the Unicode encoding (overwriting the defaults above).

```
\ifx \Umathcode\@undefined  \else
```

This set should be taken from TS1 encoding even if it means you get it from the default font for that encoding.

```
%\DeclareTextSymbol{\textcopyleft}{TS1}{171}
%\DeclareTextSymbol{\textdblhyphen}{TS1}{45}
%\DeclareTextSymbol{\textdblhyphenchar}{TS1}{127}
%\DeclareTextSymbol{\textquotestraightbase}{TS1}{13}
%\DeclareTextSymbol{\textquotestraightdblbase}{TS1}{18}
%\DeclareTextSymbol{\textleaf}{TS1}{108}
%\DeclareTextSymbol{\texttwelveudash}{TS1}{21}
%\DeclareTextSymbol{\textthreequartersemdash}{TS1}{22}
```

If oldstyle numerals are asked for we just use \oldstylenums.

```
\DeclareTextCommand{\textzerooldstyle} \UnicodeEncodingName{\oldstylenums{0}}
\DeclareTextCommand{\textoneoldstyle}  \UnicodeEncodingName{\oldstylenums{1}}
\DeclareTextCommand{\texttwooldstyle}  \UnicodeEncodingName{\oldstylenums{2}}
\DeclareTextCommand{\textthreeoldstyle}\UnicodeEncodingName{\oldstylenums{3}}
\DeclareTextCommand{\textfouroldstyle} \UnicodeEncodingName{\oldstylenums{4}}
\DeclareTextCommand{\textfiveoldstyle} \UnicodeEncodingName{\oldstylenums{5}}
```

```
\DeclareTextCommand{\textsixoldstyle}  \UnicodeEncodingName{\oldstylenums{6}}
\DeclareTextCommand{\textsevenoldstyle}\UnicodeEncodingName{\oldstylenums{7}}
\DeclareTextCommand{\texteightoldstyle}\UnicodeEncodingName{\oldstylenums{8}}
\DeclareTextCommand{\textnineoldstyle} \UnicodeEncodingName{\oldstylenums{9}}
```

These have Unicode slots so this should be integrated into TU explictly

```
\DeclareTextSymbol{\textpilcrow}       \UnicodeEncodingName{"00B6}
\DeclareTextSymbol{\textborn}          \UnicodeEncodingName{"002A}
\DeclareTextSymbol{\textdied}          \UnicodeEncodingName{"2020}
\DeclareTextSymbol{\textlbrackdbl}     \UnicodeEncodingName{"27E6}
\DeclareTextSymbol{\textrbrackdbl}     \UnicodeEncodingName{"27E7}
\DeclareTextSymbol{\textguarani}       \UnicodeEncodingName{"20B2}
```

We could make `\textcentoldstyle` and `\textdollaroldstyle` point to dollar and cent in the Unicode encoding

```
%\DeclareTextSymbol{\textcentoldstyle}           \UnicodeEncodingName{"00A2}
%\DeclareTextSymbol{\textdollaroldstyle}          \UnicodeEncodingName{"0024}
```

but I think it is better to pick them up from TS1 even if that usually means LMR fonts

```
\DeclareTextSymbol{\textdollaroldstyle}{TS1}{138}
\DeclareTextSymbol{\textcentoldstyle}  {TS1}{139}
```

```
\fi               % --- END of Unicode engines specials
```

3 Font family sub-encodings setup

We declare the subsets for a good number of fonts in the kernel ...

But first the default for anything that is not declared. We use `9` which is most likely much too conservative, but with the advantage that we aren't getting missing glyphs (or at least that this is very unlikely). For nearly all font in the TEX Live distribution of 2019 "correct" classifications are given below, so that this default is only used for new font families, and over time the right classifications can be added here too.

```
\DeclareEncodingSubset{TS1}{?}{9}
```

This first block contains the fonts that have been already supported by the `textcomp` package way back, i.e., the font families that have TEX support since the mid-nineties.

```
\DeclareEncodingSubset{TS1}{ccr}     {0}
\DeclareEncodingSubset{TS1}{cmbr}    {0}
\DeclareEncodingSubset{TS1}{cmr}     {0}
\DeclareEncodingSubset{TS1}{cmss}    {0}
\DeclareEncodingSubset{TS1}{cmtl}    {0}
\DeclareEncodingSubset{TS1}{cmtt}    {0}
\DeclareEncodingSubset{TS1}{cmvtt}   {0}
\DeclareEncodingSubset{TS1}{pxr}     {0}
\DeclareEncodingSubset{TS1}{pxss}    {0}
\DeclareEncodingSubset{TS1}{pxtt}    {0}
\DeclareEncodingSubset{TS1}{qag}     {0}
\DeclareEncodingSubset{TS1}{qbk}     {0}
\DeclareEncodingSubset{TS1}{qcr}     {0}
\DeclareEncodingSubset{TS1}{qcs}     {0}
\DeclareEncodingSubset{TS1}{qhvc}    {0}
\DeclareEncodingSubset{TS1}{qhv}     {0}
\DeclareEncodingSubset{TS1}{qpl}     {0}
\DeclareEncodingSubset{TS1}{qtm}     {0}
```

```
\DeclareEncodingSubset{TS1}{qzc}     {0}
\DeclareEncodingSubset{TS1}{txr}     {0}
\DeclareEncodingSubset{TS1}{txss}    {0}
\DeclareEncodingSubset{TS1}{txtt}    {0}

\DeclareEncodingSubset{TS1}{lmr}     {1}
\DeclareEncodingSubset{TS1}{lmdh}    {1}
\DeclareEncodingSubset{TS1}{lmss}    {1}
\DeclareEncodingSubset{TS1}{lmssq}   {1}
\DeclareEncodingSubset{TS1}{lmvtt}   {1}
\DeclareEncodingSubset{TS1}{lmtt}    {1} % missing TM, SM, pertenthousand for some reason

\DeclareEncodingSubset{TS1}{ptmx}    {2}
\DeclareEncodingSubset{TS1}{ptmj}    {2}
\DeclareEncodingSubset{TS1}{ul8}     {2}

\DeclareEncodingSubset{TS1}{bch}     {5}  % tofu for blank, ohm
\DeclareEncodingSubset{TS1}{futj}    {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{futs}    {5}  % tofu for blank, ohm
\DeclareEncodingSubset{TS1}{futx}    {5}  % probably (currently broken distrib)
\DeclareEncodingSubset{TS1}{pag}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{pbk}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{pcr}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{phv}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{pnc}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{pplj}    {5}  % tofu for blank
\DeclareEncodingSubset{TS1}{pplx}    {5}  % tofu for blank
\DeclareEncodingSubset{TS1}{ppl}     {5}  % tofu for blank interrobang/down
\DeclareEncodingSubset{TS1}{ptm}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{pzc}     {5}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{ul9}     {5}  % tofu for blank, interrobang/down, ohm

\DeclareEncodingSubset{TS1}{dayroms} {6}  % tofu for blank, interrobang/down, ohm
\DeclareEncodingSubset{TS1}{dayrom}  {6}  % tofu for blank, interrobang/down, ohm

\DeclareEncodingSubset{TS1}{augie}   {8}  % really only missing euro
\DeclareEncodingSubset{TS1}{put}     {8}
\DeclareEncodingSubset{TS1}{uag}     {8}  % probably (currently broken distrib)
\DeclareEncodingSubset{TS1}{ugq}     {8}

\DeclareEncodingSubset{TS1}{zi4}     {9}
```

LucidaBright (sold through TUG) probably not quite correct, I guess as I have the older fonts ...

```
\DeclareEncodingSubset{TS1}{hls}     {5}
\DeclareEncodingSubset{TS1}{hlst}    {5}
\DeclareEncodingSubset{TS1}{hlct}    {5}
\DeclareEncodingSubset{TS1}{hlh}     {5}
\DeclareEncodingSubset{TS1}{hlx}     {8}
\DeclareEncodingSubset{TS1}{hlce}    {8}
\DeclareEncodingSubset{TS1}{hlcn}    {8}
\DeclareEncodingSubset{TS1}{hlcw}    {8}
\DeclareEncodingSubset{TS1}{hlcf}    {8}
```

Below are the newer fonts that have support files for LaTeX. With very few exceptions the classifications are done so that all characters are correctly produced (either being available in the font or substituted.

There are a few fonts that contain “tofu” squares in places (instead of a real glyph) and in a few cases some really seldom needed chars are unavailable, i.e., produce missing glyphs (to avoid that a large number of available chars are unnecessarily substituted.

```
\DeclareEncodingSubset{TS1}{lato-*}                   {0}  % with a bunch of tofu inside
\DeclareEncodingSubset{TS1}{opensans-*}               {0}  % with a bunch of tofu inside
\DeclareEncodingSubset{TS1}{cantarell-*}              {0}  % with a bunch of tofu inside
\DeclareEncodingSubset{TS1}{fbb-*}                    {0}  % missing centoldstyle

\DeclareEncodingSubset{TS1}{tli}                      {1}  % with lots of tofu inside

\DeclareEncodingSubset{TS1}{Alegreya-*}               {2}
\DeclareEncodingSubset{TS1}{AlegreyaSans-*}           {2}
\DeclareEncodingSubset{TS1}{DejaVuSans-TLF}           {2}
\DeclareEncodingSubset{TS1}{DejaVuSansCondensed-TLF}  {2}
\DeclareEncodingSubset{TS1}{DejaVuSansMono-TLF}       {2}
\DeclareEncodingSubset{TS1}{EBGaramond-*}             {2}
\DeclareEncodingSubset{TS1}{Tempora-TLF}              {2}
\DeclareEncodingSubset{TS1}{Tempora-TOsF}             {2}

\DeclareEncodingSubset{TS1}{Arimo-TLF}                {3}
\DeclareEncodingSubset{TS1}{Carlito-*}                {3}
\DeclareEncodingSubset{TS1}{FiraSans-*}               {3}
\DeclareEncodingSubset{TS1}{IBMPlexSans-TLF}          {3}
\DeclareEncodingSubset{TS1}{Merriweather-OsF}         {3}
\DeclareEncodingSubset{TS1}{Montserrat-*}             {3}
\DeclareEncodingSubset{TS1}{MontserratAlternates-*}   {3}
\DeclareEncodingSubset{TS1}{SourceCodePro-TLF}        {3}
\DeclareEncodingSubset{TS1}{SourceCodePro-TOsF}       {3}
\DeclareEncodingSubset{TS1}{SourceSansPro-*}          {3}
\DeclareEncodingSubset{TS1}{SourceSerifPro-*}         {3}
\DeclareEncodingSubset{TS1}{Tinos-TLF}                {3}

\DeclareEncodingSubset{TS1}{AccanthisADFStdNoThree-LF}{4}
\DeclareEncodingSubset{TS1}{Cabin-TLF}                {4}
\DeclareEncodingSubset{TS1}{Caladea-TLF}              {4}
\DeclareEncodingSubset{TS1}{Chivo-*}                  {4}
\DeclareEncodingSubset{TS1}{ClearSans-TLF}            {4}
\DeclareEncodingSubset{TS1}{Coelacanth-LF}            {4}
\DeclareEncodingSubset{TS1}{CrimsonPro-*}             {4}
\DeclareEncodingSubset{TS1}{FiraMono-TLF}             {4}
\DeclareEncodingSubset{TS1}{FiraMono-TOsF}            {4}
\DeclareEncodingSubset{TS1}{Go-TLF}                   {4}
\DeclareEncodingSubset{TS1}{GoMono-TLF}               {4}
\DeclareEncodingSubset{TS1}{InriaSans-*}              {4}
\DeclareEncodingSubset{TS1}{InriaSerif-*}             {4}
\DeclareEncodingSubset{TS1}{LibertinusSans-*}         {4}
\DeclareEncodingSubset{TS1}{LibertinusSerif-*}        {4}
\DeclareEncodingSubset{TS1}{LibreBodoni-TLF}          {4}
\DeclareEncodingSubset{TS1}{LibreFranklin-TLF}        {4}
\DeclareEncodingSubset{TS1}{LinguisticsPro-LF}        {4}
\DeclareEncodingSubset{TS1}{LinguisticsPro-OsF}       {4}
\DeclareEncodingSubset{TS1}{LinuxBiolinumT-*}         {4}
\DeclareEncodingSubset{TS1}{LinuxLibertineT-*}        {4}
\DeclareEncodingSubset{TS1}{MerriweatherSans-OsF}     {4}
\DeclareEncodingSubset{TS1}{MintSpirit-*}             {4}
\DeclareEncodingSubset{TS1}{MintSpiritNoTwo-*}        {4}
```

```
\DeclareEncodingSubset{TS1}{PTMono-TLF}                {4}
\DeclareEncodingSubset{TS1}{PTSans-TLF}                {4}
\DeclareEncodingSubset{TS1}{PTSansCaption-TLF}         {4}
\DeclareEncodingSubset{TS1}{PTSansNarrow-TLF}          {4}
\DeclareEncodingSubset{TS1}{PTSerif-TLF}               {4}
\DeclareEncodingSubset{TS1}{PTSerifCaption-TLF}        {4}
\DeclareEncodingSubset{TS1}{Raleway-TLF}               {4}
\DeclareEncodingSubset{TS1}{Raleway-TOsF}              {4}
\DeclareEncodingSubset{TS1}{Roboto-*}                  {4}
\DeclareEncodingSubset{TS1}{RobotoMono-TLF}            {4}
\DeclareEncodingSubset{TS1}{RobotoSlab-TLF}            {4}
\DeclareEncodingSubset{TS1}{Rosario-*}                 {4}
\DeclareEncodingSubset{TS1}{SticksTooText-*}           {4}
\DeclareEncodingSubset{TS1}{UniversalisADFStd-LF}      {4}

\DeclareEncodingSubset{TS1}{Almendra-OsF}              {5}
\DeclareEncodingSubset{TS1}{Baskervaldx-*}             {5}
\DeclareEncodingSubset{TS1}{BaskervilleF-*}            {5}
\DeclareEncodingSubset{TS1}{Bitter-TLF}                {5}
\DeclareEncodingSubset{TS1}{Cinzel-LF}                 {5}
\DeclareEncodingSubset{TS1}{CinzelDecorative-LF}       {5}
\DeclareEncodingSubset{TS1}{DejaVuSerif-TLF}           {5}
\DeclareEncodingSubset{TS1}{DejaVuSerifCondensed-TLF} {5}
\DeclareEncodingSubset{TS1}{GilliusADF-LF}             {5}
\DeclareEncodingSubset{TS1}{GilliusADFCond-LF}         {5}
\DeclareEncodingSubset{TS1}{GilliusADFNoTwo-LF}        {5}
\DeclareEncodingSubset{TS1}{GilliusADFNoTwoCond-LF}    {5}
\DeclareEncodingSubset{TS1}{LobsterTwo-LF}             {5}
\DeclareEncodingSubset{TS1}{OldStandard-TLF}           {5}
\DeclareEncodingSubset{TS1}{PlayfairDisplay-TLF}       {5}
\DeclareEncodingSubset{TS1}{PlayfairDisplay-TOsF}      {5}
\DeclareEncodingSubset{TS1}{TheanoDidot-TLF}           {5}
\DeclareEncodingSubset{TS1}{TheanoDidot-TOsF}          {5}
\DeclareEncodingSubset{TS1}{TheanoModern-TLF}          {5}
\DeclareEncodingSubset{TS1}{TheanoModern-TOsF}         {5}
\DeclareEncodingSubset{TS1}{TheanoOldStyle-TLF}        {5}
\DeclareEncodingSubset{TS1}{TheanoOldStyle-TOsF}       {5}

\DeclareEncodingSubset{TS1}{Crimson-TLF}               {6}
\DeclareEncodingSubset{TS1}{IBMPlexMono-TLF}           {6}
\DeclareEncodingSubset{TS1}{IBMPlexSerif-TLF}          {6}
\DeclareEncodingSubset{TS1}{LibertinusMono-TLF}        {6}
\DeclareEncodingSubset{TS1}{LibertinusSerifDisplay-LF}{6}
\DeclareEncodingSubset{TS1}{LinuxLibertineDisplayT-*} {6}
\DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-LF}   {6}
\DeclareEncodingSubset{TS1}{LinuxLibertineMonoT-TLF}  {6}
\DeclareEncodingSubset{TS1}{Overlock-LF}               {6}

\DeclareEncodingSubset{TS1}{CormorantGaramond-*}       {7}
\DeclareEncodingSubset{TS1}{Heuristica-TLF}            {7}
\DeclareEncodingSubset{TS1}{Heuristica-TOsF}           {7}
\DeclareEncodingSubset{TS1}{IMFELLEnglish-TLF}         {7}
\DeclareEncodingSubset{TS1}{LibreBaskerville-TLF}      {7}
\DeclareEncodingSubset{TS1}{LibreCaslon-*}             {7}
\DeclareEncodingSubset{TS1}{Marcellus-LF}              {7}
\DeclareEncodingSubset{TS1}{NotoSans-*}                {7}
```

```
\DeclareEncodingSubset{TS1}{NotoSansMono-TLF}         {7}
\DeclareEncodingSubset{TS1}{NotoSansMono-TOsF}        {7}
\DeclareEncodingSubset{TS1}{NotoSerif-*}              {7}
\DeclareEncodingSubset{TS1}{Quattrocento-TLF}         {7}
\DeclareEncodingSubset{TS1}{QuattrocentoSans-TLF}     {7}
\DeclareEncodingSubset{TS1}{XCharter-TLF}             {7}
\DeclareEncodingSubset{TS1}{XCharter-TOsF}            {7}
\DeclareEncodingSubset{TS1}{erewhon-*}                {7}
\DeclareEncodingSubset{TS1}{ComicNeue-TLF}            {7}
\DeclareEncodingSubset{TS1}{ComicNeueAngular-TLF}     {7}
\DeclareEncodingSubset{TS1}{Forum-LF}                 {7}  % the superiors are missing
\DeclareEncodingSubset{TS1}{Cochineal-*}              {8}
\DeclareEncodingSubset{TS1}{AlgolRevived-TLF}         {9}
```

4 Legacy symbol support for lists and footnote symbols

\UseLegacyTextSymbols

```
\def\UseLegacyTextSymbols{%
  \DeclareTextSymbolDefault{\textasteriskcentered}{OMS}%
  \DeclareTextSymbolDefault{\textbardbl}{OMS}%
  \DeclareTextSymbolDefault{\textbullet}{OMS}%
  \DeclareTextSymbolDefault{\textdaggerdbl}{OMS}%
  \DeclareTextSymbolDefault{\textdagger}{OMS}%
  \DeclareTextSymbolDefault{\textparagraph}{OMS}%
  \DeclareTextSymbolDefault{\textperiodcentered}{OMS}%
  \DeclareTextSymbolDefault{\textsection}{OMS}%
  \UndeclareTextCommand{\textsection}{T1}%
  \expandafter\let\csname oldstylenums \expandafter\endcsname
                  \csname legacyoldstylenums \endcsname
}
```

(*End definition for* `\UseLegacyTextSymbols`*. This function is documented on page* **??**.)

\textlegacyasteriskcentered
\textlegacybardbl
\textlegacybullet
\textlegacydaggerdbl
\textlegacydagger
\textlegacyparagraph
\textlegacyperiodcentered
\textlegacysection

Here are new names for the legacy symbols that LaTeX used to pick up from the OMS encoded fonts (and used dor itemize lists or footnote symbols.

We go the roundabout way via separate OMS declarations so that

```
\renewcommand\textbullet{\textlegacybullet}
```

doesn't produce an endless loop.

```
\DeclareTextSymbol{\textlegacyasteriskcentered}{OMS}{3}   % "03
\DeclareTextSymbol{\textlegacybardbl}{OMS}{107}           % "6B
\DeclareTextSymbol{\textlegacybullet}{OMS}{15}            % "0F
\DeclareTextSymbol{\textlegacydaggerdbl}{OMS}{122}        % "7A
\DeclareTextSymbol{\textlegacydagger}{OMS}{121}           % "79
\DeclareTextSymbol{\textlegacyparagraph}{OMS}{123}        % "7B
\DeclareTextSymbol{\textlegacyperiodcentered}{OMS}{1}     % "01
\DeclareTextSymbol{\textlegacysection}{OMS}{120}          % "78
```

```
\DeclareTextSymbolDefault{\textlegacyasteriskcentered}{OMS}
\DeclareTextSymbolDefault{\textlegacybardbl}{OMS}
\DeclareTextSymbolDefault{\textlegacybullet}{OMS}
\DeclareTextSymbolDefault{\textlegacydaggerdbl}{OMS}
\DeclareTextSymbolDefault{\textlegacydagger}{OMS}
\DeclareTextSymbolDefault{\textlegacyparagraph}{OMS}
\DeclareTextSymbolDefault{\textlegacyperiodcentered}{OMS}
\DeclareTextSymbolDefault{\textlegacysection}{OMS}
```

(*End definition for* `\textlegacyasteriskcentered` *and others. These functions are documented on page* ??.)

Supporting rollback ...

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩    {\DeclareEncodingSubset}{Text companion symbols}%
⟨latexrelease⟩
⟨latexrelease⟩\let\DeclareEncodingSubset\@undefined
⟨latexrelease⟩\let\CheckEncodingSubset\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareTextSymbolDefault{\textdollar}{OT1}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textsterling}{OT1}
⟨latexrelease⟩\DeclareTextCommand{\textdollar}{OT1}{\hmode@bgroup
⟨latexrelease⟩    \ifdim \fontdimen\@ne\font >\z@
⟨latexrelease⟩       \slshape
⟨latexrelease⟩    \else
⟨latexrelease⟩       \upshape
⟨latexrelease⟩    \fi
⟨latexrelease⟩    \char`\$\egroup}
⟨latexrelease⟩\DeclareTextCommand{\textsterling}{OT1}{\hmode@bgroup
⟨latexrelease⟩    \ifdim \fontdimen\@ne\font >\z@
⟨latexrelease⟩       \itshape
⟨latexrelease⟩    \else
⟨latexrelease⟩       \fontshape{ui}\selectfont
⟨latexrelease⟩    \fi
⟨latexrelease⟩    \char`\$\egroup}
⟨latexrelease⟩\DeclareTextCommand{\textperthousand}{T1}
⟨latexrelease⟩    {\%\char 24 }
⟨latexrelease⟩
⟨latexrelease⟩\DeclareTextSymbolDefault{\textasteriskcentered}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textbullet}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textdaggerdbl}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textdagger}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textparagraph}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textperiodcentered}{OMS}
⟨latexrelease⟩\DeclareTextSymbolDefault{\textsection}{OMS}
⟨latexrelease⟩
⟨latexrelease⟩\DeclareTextSymbolDefault{\textbardbl}{OMS}
⟨latexrelease⟩\let\textbrokenbar\@undefined
⟨latexrelease⟩\let\textcelsius\@undefined
⟨latexrelease⟩\let\textcent\@undefined
⟨latexrelease⟩\DeclareTextCommandDefault{\textcopyright}{\textcircled{c}}
⟨latexrelease⟩\let\textdegree\@undefined
⟨latexrelease⟩\let\textdiv\@undefined
```

```
⟨latexrelease⟩\let\textlnot\@undefined
⟨latexrelease⟩\let\textonehalf\@undefined
⟨latexrelease⟩\let\textonequarter\@undefined
⟨latexrelease⟩\let\textonesuperior\@undefined
⟨latexrelease⟩\DeclareTextCommandDefault{\textordfeminine}{\textsuperscript{a}}
⟨latexrelease⟩\DeclareTextCommandDefault{\textordmasculine}{\textsuperscript{o}}
⟨latexrelease⟩\let\textpm\@undefined
⟨latexrelease⟩\let\textquotesingle\@undefined
⟨latexrelease⟩\let\textquotestraightbase\@undefined
⟨latexrelease⟩\let\textquotestraightdblbase\@undefined
⟨latexrelease⟩\DeclareTextCommandDefault{\textregistered}{\textcircled{%
⟨latexrelease⟩      \check@mathfonts\fontsize\sf@size\z@\math@fontsfalse\selectfont R}}
⟨latexrelease⟩\let\textthreequartersemdash\@undefined
⟨latexrelease⟩\let\textthreequarters\@undefined
⟨latexrelease⟩\let\textthreesuperior\@undefined
⟨latexrelease⟩\let\texttimes\@undefined
⟨latexrelease⟩\DeclareTextCommandDefault{\texttrademark}{\textsuperscript{TM}}
⟨latexrelease⟩\let\texttwelveudash\@undefined
⟨latexrelease⟩\let\texttwosuperior\@undefined
⟨latexrelease⟩\let\textyen\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textcapitalcompwordmark\@undefined
⟨latexrelease⟩\let\textascendercompwordmark\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareTextAccentDefault{\textcircled}{OMS}
⟨latexrelease⟩\DeclareTextAccentDefault{\t}{OML}
⟨latexrelease⟩
⟨latexrelease⟩\let\capitalacute\@undefined
⟨latexrelease⟩\let\capitalbreve\@undefined
⟨latexrelease⟩\let\capitalcaron\@undefined
⟨latexrelease⟩\let\capitalcedilla\@undefined
⟨latexrelease⟩\let\capitalcircumflex\@undefined
⟨latexrelease⟩\let\capitaldieresis\@undefined
⟨latexrelease⟩\let\capitaldotaccent\@undefined
⟨latexrelease⟩\let\capitalgrave\@undefined
⟨latexrelease⟩\let\capitalhungarumlaut\@undefined
⟨latexrelease⟩\let\capitalmacron\@undefined
⟨latexrelease⟩\let\capitalnewtie\@undefined
⟨latexrelease⟩\let\capitalogonek\@undefined
⟨latexrelease⟩\let\capitalring\@undefined
⟨latexrelease⟩\let\capitaltie\@undefined
⟨latexrelease⟩\let\capitaltilde\@undefined
⟨latexrelease⟩\let\newtie\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textlbrackdbl\@undefined
⟨latexrelease⟩\let\textrbrackdbl\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\texteightoldstyle\@undefined
⟨latexrelease⟩\let\textfiveoldstyle\@undefined
⟨latexrelease⟩\let\textfouroldstyle\@undefined
⟨latexrelease⟩\let\textnineoldstyle\@undefined
⟨latexrelease⟩\let\textoneoldstyle\@undefined
⟨latexrelease⟩\let\textsevenoldstyle\@undefined
⟨latexrelease⟩\let\textsixoldstyle\@undefined
```

```
⟨latexrelease⟩\let\textthreeoldstyle\@undefined
⟨latexrelease⟩\let\texttwooldstyle\@undefined
⟨latexrelease⟩\let\textzerooldstyle\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textacutedbl\@undefined
⟨latexrelease⟩\let\textasciiacute\@undefined
⟨latexrelease⟩\let\textasciibreve\@undefined
⟨latexrelease⟩\let\textasciicaron\@undefined
⟨latexrelease⟩\let\textasciidieresis\@undefined
⟨latexrelease⟩\let\textasciigrave\@undefined
⟨latexrelease⟩\let\textasciimacron\@undefined
⟨latexrelease⟩\let\textgravedbl\@undefined
⟨latexrelease⟩\let\texttildelow\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textbaht\@undefined
⟨latexrelease⟩\let\textbigcircle\@undefined
⟨latexrelease⟩\let\textborn\@undefined
⟨latexrelease⟩\let\textcentoldstyle\@undefined
⟨latexrelease⟩\let\textcircledP\@undefined
⟨latexrelease⟩\let\textcopyleft\@undefined
⟨latexrelease⟩\let\textdblhyphenchar\@undefined
⟨latexrelease⟩\let\textdblhyphen\@undefined
⟨latexrelease⟩\let\textdied\@undefined
⟨latexrelease⟩\let\textdiscount\@undefined
⟨latexrelease⟩\let\textdivorced\@undefined
⟨latexrelease⟩\let\textdollaroldstyle\@undefined
⟨latexrelease⟩\let\textguarani\@undefined
⟨latexrelease⟩\let\textleaf\@undefined
⟨latexrelease⟩\let\textlquill\@undefined
⟨latexrelease⟩\let\textmarried\@undefined
⟨latexrelease⟩\let\textmho\@undefined
⟨latexrelease⟩\let\textmusicalnote\@undefined
⟨latexrelease⟩\let\textnaira\@undefined
⟨latexrelease⟩\let\textopenbullet\@undefined
⟨latexrelease⟩\let\textpeso\@undefined
⟨latexrelease⟩\let\textpilcrow\@undefined
⟨latexrelease⟩\let\textrecipe\@undefined
⟨latexrelease⟩\let\textreferencemark\@undefined
⟨latexrelease⟩\let\textrquill\@undefined
⟨latexrelease⟩\let\textservicemark\@undefined
⟨latexrelease⟩\let\textsurd\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareTextCommand{\textpertenthousand}{T1}
⟨latexrelease⟩                   {\%\char 24\char 24 }
⟨latexrelease⟩
⟨latexrelease⟩\let\textlangle\@undefined
⟨latexrelease⟩\let\textrangle\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textcolonmonetary\@undefined
⟨latexrelease⟩\let\textdong\@undefined
⟨latexrelease⟩\let\textdownarrow\@undefined
⟨latexrelease⟩\let\textleftarrow\@undefined
⟨latexrelease⟩\let\textlira\@undefined
⟨latexrelease⟩\let\textrightarrow\@undefined
```

```
⟨latexrelease⟩\let\textuparrow\@undefined
⟨latexrelease⟩\let\textwon\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textestimated\@undefined
⟨latexrelease⟩\let\textnumero\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textflorin\@undefined
⟨latexrelease⟩\let\textcurrency\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textfractionsolidus\@undefined
⟨latexrelease⟩\let\textohm\@undefined
⟨latexrelease⟩\let\textmu\@undefined
⟨latexrelease⟩\let\textminus\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textblank\@undefined
⟨latexrelease⟩\let\textinterrobangdown\@undefined
⟨latexrelease⟩\let\textinterrobang\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\texteuro\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\textcelsius\@undefined
⟨latexrelease⟩\let\textonesuperior\@undefined
⟨latexrelease⟩\let\textthreequartersemdash\@undefined
⟨latexrelease⟩\let\textthreesuperior\@undefined
⟨latexrelease⟩\let\texttwelveudash\@undefined
⟨latexrelease⟩\let\texttwosuperior\@undefined
⟨latexrelease⟩\let\textbardbl\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let\UseLegacyTextSymbols\@undefined
⟨latexrelease⟩\let\textlegacyasteriskcentered\@undefined
⟨latexrelease⟩\let\textlegacybardbl\@undefined
⟨latexrelease⟩\let\textlegacybullet\@undefined
⟨latexrelease⟩\let\textlegacydaggerdbl\@undefined
⟨latexrelease⟩\let\textlegacydagger\@undefined
⟨latexrelease⟩\let\textlegacyparagraph\@undefined
⟨latexrelease⟩\let\textlegacyperiodcentered\@undefined
⟨latexrelease⟩\let\textlegacysection\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
⟨/2ekernel⟩
```

5 The textcomp package

```
⟨*TS1sty⟩
\providecommand\DeclareRelease[3]{}
\providecommand\DeclareCurrentRelease[2]{}

\DeclareRelease{}{2018-08-11}{textcomp-2018-08-11.sty}
\DeclareCurrentRelease{}{2020-02-02}

\ProvidesPackage{textcomp}
 [2020/02/02 v2.0n Standard LaTeX package]
```

A precaution in case this is used without rebuilding the format.

```
\NeedsTeXFormat{LaTeX2e}[2020/02/02]
```

This is implemented by defining the default subset:

```
\DeclareOption{full}{\DeclareEncodingSubset{TS1}{?}{0}}
\DeclareOption{almostfull}{\DeclareEncodingSubset{TS1}{?}{1}}
\DeclareOption{euro}{\DeclareEncodingSubset{TS1}{?}{8}}
\DeclareOption{safe}{\DeclareEncodingSubset{TS1}{?}{9}}
```

The default is set up in the kernel is "safe" these days for unknown fonts but LaTeX has definitions for most families so it seldom applies.

If a different default is used then one needs to check the results to ensure that there aren't "missing glyphs".

The next set of options define the warning level (default in the kernel is info only). Using the package options you can change this behavior.

```
\DeclareOption{error}{\gdef\tc@errorwarn{\PackageError{textcomp}}}
\DeclareOption{warn}{\gdef\tc@errorwarn#1#2{\PackageWarning{textcomp}{#1}}}
\DeclareOption{info}{\gdef\tc@errorwarn#1#2{\PackageInfo{textcomp}{#1}}}
\DeclareOption{quiet}{\gdef\tc@errorwarn#1#2{}}
```

The "force" option basically changes the sub-encoding to that of the default (which, unless changes, is 9 these days), i.e., it no longer depends on the font in use. This is mainly there because it might have been used in older documents, but not somehting that is recommended.

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

```
\ExecuteOptions{info}
\ProcessOptions\relax
```

There is not much else to do nowadays, because everything is already set up in the LaTeX kernel.

```
\InputIfFileExists{textcomp.cfg}
  {\PackageInfo{textcomp}{Local configuration file used}}{}
```

⟨/TS1sty⟩

5.1 The old textcomp package code

This section contains the old code for the textcomp package and its documentation. It is only used if we roll back prior to 2020. Thus all the rest is mainly for historians. Note that the old code categorised in the sub-encodings only into 6 classes not 10.

⟨*TS1oldsty⟩

```
\ProvidesPackage{textcomp}
   [2018/08/11 v2.0j Standard LaTeX package]
```

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

Since many PostScript fonts only implement a subset of TS1 many commands only produce black blobs of ink. To resolve the resulting problems a number of options have been introduced and some code has been developed to distinguish sub-encodings.

The sub-encodings have a numerical id and are defined as follows for TS1:

#5 those TS1 symbols that are also in the ISO-Adobe character set; without textcurrency, which is often misused for the Euro. Older Type1 fonts from the non-TEX world provide only this subset.

#4 = #5 + \texteuro. Most newer fonts provide this.

#3 = #4 + \textomega. Can also be described as TS1 ∩ (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 the command \DeclareEncodingSubset that takes the major encoding as the first argument (e.g., TS1), the family name as the second argument (e.g., cmr), and the subset encoding id as a third, (e.g., 0 for cmr).

The default encoding subset to use when nothing is known about the current font family is named ?.

```
\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
```

(*End definition for* `\DeclareEncodingSubset`*. This function is documented on page* **??**.)

The options for the package are the following:

safe for unknown font families enables only symbols that are also in the ISO-Adobe character set; without "currency", which is often misused for the Euro. Older Type1 fonts from the non-TeX world provide only this subset.

euro enables the "safe" symbols plus the `\texteuro` command. Most newer fonts provide this.

full enables all `TS1` commands; useful only with fonts like EC or CM bright.

almostfull same as "full", except that `\textcircled` and `\t` are *not* redefined from their defaults to avoid that commands like `\copyright` suddenly no longer work.

force ignore all subset encoding definitions stored in the package itself or in the configuration file and always use the default subset as specified by one of the other options (seldom useful, only dangerous).

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

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

(*End definition for* `\iftc@forced`*. This function is documented on page* **??**.)

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}}}
\DeclareOption{quiet}{\gdef\tc@errorwarn#1#2#3{}}
```

```
\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
```

(*End definition for* `\CheckEncodingSubset`*. This function is documented on page* **??**.)

`\tc@subst`

```
\def\tc@subst#1{%
   \tc@errorwarn{textcomp}%
    {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
}
```

(*End definition for* `\tc@subst`*. This function is documented on page* ??.)

`\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
}
```

(*End definition for* `\tc@error`*. This function is documented on page* ??.)

`\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
   }%
}
```

(*End definition for* `\tc@fake@euro`*. This function is documented on page* ??.)

`\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}
```

(*End definition for* `\tc@check@symbol` *and* `\tc@check@accent`*. These functions are documented on page* ??.)

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 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).

```
\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. 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}
```

5.1.1 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
}
```

5.1.2 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}{lmtt}    {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
```

⟨/TS1oldsty⟩

File B
ltpageno.dtx

1 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 C
ltxref.dtx

1 Cross Referencing

The user writes `\label{⟨foo⟩}` to define the following cross-references:

`\ref{⟨foo⟩}`: value of most recently incremented referencable counter. in the current environment. (Chapter, section, theorem and enumeration counters are referencable, footnote counters are not.)

`\pageref{⟨foo⟩}`: page number at which `\label{foo}` command appeared. where foo can be any string of characters not containing '`\`', '`{`' or '`}`'.

Note: The scope of the `\label` command is delimited by environments, so
`\begin{theorem} \label{foo} ... \end{theorem} \label{bar}`
defines `\ref{foo}` to be the theorem number and `\ref{bar}` to be the current section number.

Note: `\label` does the right thing in terms of spacing – i.e., leaving a space on both sides of it is equivalent to leaving a space on either side.

1.1 Cross Referencing

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

```
⟨*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
```

End of historical LaTeX 2.09 comments.

`\labelformat` A reference via `\ref` produces by default the data associated with the corresponding `\label` command (typically a number); any additional formatting has to be provided by the user. If, for example, references to equations are always to be typeset as "equation (*number*)", one has to code "`equation (\ref{`*key*`})`". With `\labelformat` there is a possibility to generate such frills automatically without resorting to low-level coding. The command takes two arguments: the first is the name of a counter and the second is its representation when referenced. This means that for a successful usage, one has to know the counter name being used for generating the label, though in practice this should not pose a problem. The current counter number is picked up as an argument. Here are two examples:

```
\labelformat{section}{section~#1}
\labelformat{equation}{equation~(#1)}}
```

`\Ref` A side effect of using `\labelformat` is that, depending on the defined formatting, it becomes impossible to use `\ref` at the beginning of a sentence (if its replacement text starts with a lowercase letter). To overcome this problem we introduce the command `\Ref` that behave like `\ref` except that it uppercases the first token of the generated string.

To make `\Ref` work properly the very first token in the second argument of `\labelformat` has to be a simple ASCII or UTF-8 letter, otherwise the capitalization will fail or worse, you will end up with some error messages. If you actually need something more complicated in this place (e.g., an accented letter not written as a UTF-8 character) you have to explicitly surround it with braces, to identify the part that needs to be capitalized. For example, for figure references in the Hungarian language you might want to write `\labelformat{figure}{{\'a}bra~\thefigure}` or use `\labelformat{figure}{ábra~\thefigure}` which avoids the brace problem.

`\G@refundefinedtrue` `\@refundefined` This does not save on name-space (since `\G@refundefinedfalse` was never needed) but it does make the implementation of such one-way switches more consistent. The extra macro to make the change is used since this change appears several times.

Note despite its name, `\G@refundefinedtrue` does *not* correspond to an `\if` command, and there is no matching `...false`. It would be more natural to call the command `\G@refundefined` (as inspection of the change log will reveal) but unfortunately such a change would break any package that had defined a `\ref`-like command that mimicked the definition of `\ref`, calling `\G@refundefinedtrue`. Inspection of the 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
```

(*End definition for* `\G@refundefinedtrue` *and* `\@refundefined`*. These functions are documented on page* ??.)

\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}}
```

(*End definition for* \ref*,* \pageref*, and* \@setref*. These functions are documented on page* ??.)

\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
```

(*End definition for* \newlabel *and* \@newl@bel*. These functions are documented on page* ??.)

\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
```

(*End definition for* \if@multiplelabels *and* \@multiplelabels*. These functions are documented on page* ??.)

\label

The commands \label and \refstepcounter have been changed to allow \protect'ed commands to work properly. For example,

```
\def\thechapter{\protect\foo{\arabic{chapter}.\roman{section}}}
```

will cause a \label{bar} command to define \ref{bar} to expand to something like \foo{4.d}. Change made 20 Jul 88.

```
\def\label#1{\@bsphack
  \protected@write\@auxout{}%
        {\string\newlabel{#1}{{\@currentlabel}{\thepage}}}%
  \@esphack}
```

(*End definition for* `\label`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\refstepcounter}{Add \@currentcounter}%
```

`\refstepcounter` Step the counter and allow for labels to point to its current value.

```
\def\@currentcounter{}
\def\refstepcounter#1{\stepcounter{#1}%
    \edef\@currentcounter{#1}%
    \protected@edef\@currentlabel
```

By generating the second csname first the `\p@...` command can grab it as an argument which can be helpful for more complicated typesetting arrangements.

The trick is to ensure that `\csname the#1\endcsname` is turned into a single token before `\p@...` is expanded further. This way, if the `\p@...` command is a macro with one argument it will receive `\the...`. With the original kernel code (i.e., without the `\expandafter`) it will instead pick up `\csname` which would be disastrous.

Using `\expandafter` instead of braces delimiting the argument is better because, assuming that the `\p@...` command is not defined as a macro with one argument, the braces will stay and prohibit kerning that might otherwise happen between the glyphs generated by `\the...` and surrounding glyphs.

```
        {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
}
```

(*End definition for* `\refstepcounter`*. This function is documented on page* **??**.)

`\labelformat` A shortcut to set the `\p@...` macro for a counter. It will pick up the counter representation as an argument so that it can be specially formatted.

```
\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
```

(*End definition for* `\labelformat`*. This function is documented on page* **??**.)

`\Ref` This macro expands the result of `\ref` and then uppercases the first token. Only useful if the label was generated via `\labelformat` and contains some lower case letter at its start. If the label starts with a complicated construct (e.g., an accented letter that is provided via a command, e.g., `\"a` instead of a UTF-8 character like ä) one has to surround everything that needs uppercasing in a brace group in the definition of `\labelformat`.[17]

```
\DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%
   \expandafter\MakeUppercase\@tempa}
```

(*End definition for* `\Ref`*. This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                    {\refstepcounter}{Add \labelformat and \Ref}%
⟨latexrelease⟩\let\@currentcounter\@undefined
⟨latexrelease⟩\def\refstepcounter#1{\stepcounter{#1}%
⟨latexrelease⟩     \protected@edef\@currentlabel
```

[17] There is one problem with this approach: the braces are kept in a normal `\ref` which might spoil kerning. Perhaps one day this needs redoing.

```
⟨latexrelease⟩      {\csname p@#1\expandafter\endcsname\csname the#1\endcsname}%
⟨latexrelease⟩}
⟨latexrelease⟩\def\labelformat#1{\expandafter\def\csname p@#1\endcsname##1}
⟨latexrelease⟩\DeclareRobustCommand\Ref[1]{\protected@edef\@tempa{\ref{#1}}%
⟨latexrelease⟩   \expandafter\MakeUppercase\@tempa}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\refstepcounter}{Add \labelformat and \Ref}%
⟨latexrelease⟩
⟨latexrelease⟩\def\refstepcounter#1{\stepcounter{#1}%
⟨latexrelease⟩    \protected@edef\@currentlabel
⟨latexrelease⟩       {\csname p@#1\endcsname\csname the#1\endcsname}%
⟨latexrelease⟩}
⟨latexrelease⟩\let\labelformat\@undefined
⟨latexrelease⟩\let\Ref\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\@currentlabel` Default for `\label` commands that come before any environment.

```
\def\@currentlabel{}
```

(*End definition for* `\@currentlabel`*. This function is documented on page* ??.)

```
⟨/2ekernel⟩
```

File D
ltmiscen.dtx

1 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,}
```

1.1 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.

Historical LATEX 2.09 comments (not necessarily accurate any more):

```
\enddocument ==
  BEGIN
   \@checkend{document}   %% checks for unmatched \begin
   \clearpage
   \begingroup
     if @filesw = true
       then  close file @mainaux
             if G@refundefined = true
              then LaTeX Warning: 'There are undefined references.' fi
             if @multiplelabels = true
               then LaTeX Warning:
                    'One or more label(s) multiply defined.'
               else
               \@setckpt {ARG1}{ARG2} == null
               \newlabel{LABEL}{VAL} ==
                    BEGIN
                      \reserved@a == VAL
                      if def(\reserved@a) = def(\r@LABEL)
                        else @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
```

End of historical LaTeX 2.09 comments.

`\@currenvir` The name of the current environment. Initialized to `document` to so that `\end{document}` works correctly.

```
\def\@currenvir{document}
```

(*End definition for* `\@currenvir`*. This function is documented on page* **??**.)

`\if@ignore`
`\@ignoretrue`
`\@ignorefalse`

```
\def\@ignorefalse{\global\let\if@ignore\iffalse}
\def\@ignoretrue {\global\let\if@ignore\iftrue}
\@ignorefalse
```

(*End definition for* `\if@ignore`*,* `\@ignoretrue`*, and* `\@ignorefalse`*. These functions are documented on page* **??**.)

`\ignorespacesafterend`

```
\let\ignorespacesafterend\@ignoretrue
```

(*End definition for* `\ignorespacesafterend`*. This function is documented on page* **??**.)

`\enddocument`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\enddocument}{Use Hooks}%
\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...

```
    \UseOneTimeHook{enddocument}%
    \@kernel@after@enddocument
```

```
  \@checkend{document}%
  \clearpage
  \UseOneTimeHook{enddocument/afterlastpage}%
  \@kernel@after@enddocument@afterlastpage
  \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
    \UseOneTimeHook{enddocument/afteraux}%
```

Next hook is expect to contain only code for writing info messages on the terminal.

```
    \UseOneTimeHook{enddocument/info}%
  \endgroup
  \UseOneTimeHook{enddocument/end}%
  \deadcycles\z@\@@end}
```

The public hooks used in `\enddocument`:

```
\NewHook{enddocument}
\NewHook{enddocument/afterlastpage}
\NewHook{enddocument/afteraux}
\NewHook{enddocument/info}
\NewHook{enddocument/end}
```

This is one of the few places where we already add data and rules to a hook already in the kernel.

```
\AddToHook{enddocument/info}[kernel/filelist]{\@dofilelist}
\AddToHook{enddocument/info}[kernel/warnings]{\@enddocument@kernel@warnings}
\DeclareHookRule{enddocument/info}{kernel/filelist}{before}{kernel/warnings}
```

(*End definition for* `\enddocument`. *This function is documented on page* **??**.)

`\@kernel@after@enddocument`
`\@kernel@after@enddocument@afterlastpage`

The two kernel hooks above are used by the shipout code.

```
\let\@kernel@after@enddocument\@empty
\let\@kernel@after@enddocument@afterlastpage\@empty
```

(*End definition for* `\@kernel@after@enddocument` *and* `\@kernel@after@enddocument@afterlastpage`. *These functions are documented on page* **??**.)

`\@enddocument@kernel@warnings`

```
\def\@enddocument@kernel@warnings{%
```

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
    \ifx \@extra@page@added \relax
        \@latex@warning@no@line{Temporary extra page added at the end.
            Rerun to get it removed}%
    \fi
```

We could think of adding a warning that nothing can be corrected while `\nofiles` is in force. In the past the warnings related to the `aux` file are simply suppressed in this case.

```
  \fi
}
```

(*End definition for* `\@enddocument@kernel@warnings`. *This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\enddocument}{Use Hooks}%
⟨latexrelease⟩
⟨latexrelease⟩\def\enddocument{%
⟨latexrelease⟩   \let\AtEndDocument\@firstofone
⟨latexrelease⟩   \@enddocumenthook
⟨latexrelease⟩   \@checkend{document}%
⟨latexrelease⟩   \clearpage
⟨latexrelease⟩   \begingroup
⟨latexrelease⟩     \if@filesw
⟨latexrelease⟩       \immediate\closeout\@mainaux
⟨latexrelease⟩       \let\@setckpt\@gobbletwo
⟨latexrelease⟩       \let\@newl@bel\@testdef
⟨latexrelease⟩       \@tempswafalse
⟨latexrelease⟩       \makeatletter \@@input\jobname.aux
```

```
⟨latexrelease⟩      \fi
⟨latexrelease⟩      \@dofilelist
⟨latexrelease⟩      \ifdim \font@submax >\fontsubfuzz\relax
⟨latexrelease⟩        \@font@warning{Size substitutions with differences\MessageBreak
⟨latexrelease⟩                   up to \font@submax\space have occurred.\@gobbletwo}%
⟨latexrelease⟩      \fi
⟨latexrelease⟩      \@defaultsubs
⟨latexrelease⟩      \@refundefined
⟨latexrelease⟩      \if@filesw
⟨latexrelease⟩        \ifx \@multiplelabels \relax
⟨latexrelease⟩          \if@tempswa
⟨latexrelease⟩            \@latex@warning@no@line{Label(s) may have changed.
⟨latexrelease⟩                Rerun to get cross-references right}%
⟨latexrelease⟩          \fi
⟨latexrelease⟩        \else
⟨latexrelease⟩          \@multiplelabels
⟨latexrelease⟩        \fi
⟨latexrelease⟩      \fi
⟨latexrelease⟩    \endgroup
⟨latexrelease⟩    \deadcycles\z@\@@end}
⟨latexrelease⟩
⟨latexrelease⟩\let\@kernel@after@enddocument\@undefined
⟨latexrelease⟩\let\@kernel@after@enddocument@afterlastpage\@undefined
⟨latexrelease⟩\let\@enddocument@kernel@warnings\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\@testdef`

```
\def\@testdef #1#2#3{%
  \def\reserved@a{#3}\expandafter \ifx \csname #1@#2\endcsname
 \reserved@a  \else \@tempswatrue \fi}
```

(*End definition for* `\@testdef`*. This function is documented on page* ??.)

Reading data from auxiliary files (like `.toc` normally happens in vertical mode and it therefore doesn't matter if line endings are converted to spaces by TEX during that process.

However, especially the `.toc` file might be read in L-R mode (in cases the `\tableofcontents` attempts to put, say a list of sub-sections as a paragraph. In that case the newlines after a line like

```
\contentsline {subsubsection}{\numberline {1.1.1}A C-head}{2}
```

might result in spurious spaces (e.g., when that level is not included).

That could be fixed by reading in the file using `\endlinechar=-1` but that has the danger that it drops some valid endlines that should be converted to spaces (for example when the user edited the TOC and then used `\nofiles` to preserve it.

So the approach taken instead is this:

- `\addcontentsline` adds the command `\protected@file@percent` to the end of the second argument of `\@writefile` that is written to the `.aux`. As the name indicates this is a protected macro so it doesn't change if it is written out.

- When the `.aux` is read back in at the end of the run, `\@writefile` is executed and writes its second argument unmodified to the file with the extension given by its first argument. Or rather that was how it was in the past.
- Instead we change `\@writefile` slightly: basically it looks at the second argument and if the last token in there is `\protected@file@percent` then it is replaced by a percent character and that is then written out. If not (for example, if the data came from a user issued `\addtocontents`, or from some package that uses `\@writefile` for writing its own files) then the command behaves exactly as before.

`\protected@file@percent` Dummy cs to be replaced by a percent sign inside `\@writefile`. If it survives (when used incorrectly) it will expand to nothing in a typsetting context.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\protected@file@percent}{Mask line endings}%
\protected\def\protected@file@percent{}
```

(*End definition for* `\protected@file@percent`*. This function is documented on page* **??**.)

`\add@percent@to@temptokena` Helper function which is used to inspect a sequence of tokens (the second argument of `\@writefile` and it the last token is `\protected@file@percent` it will replace it by a harmless percent. The result is saved in `\@temptokena` for later use.

```
\catcode`\^^A=9
\long\gdef\add@percent@to@temptokena
    #1\protected@file@percent#2\add@percent@to@temptokena
```

When we call this macro in `\@writefile` we stick in `\@empty` at the beginning, so that in case the tokenlist consists of a single brace group the braces aren't stripped. The `\expandafter` then expands this extra token away again.

```
    {\expandafter\ifx\expandafter X\detokenize{#2}X\expandafter\dont@add@percent@to@temptoken
            \expandafter\do@add@percent@to@temptokena\fi{#1}}
```

```
\long\def\dont@add@percent@to@temptokena#1{%
  \@temptokena\expandafter{#1}}
```

`latexrelease` will read this code in high-speed mode in certain situations. During that it will only look for `\if` tests but not actually execute the `\catcode` change above. As a result it will drop anything after the `%` character in the definition. Therefore the `\fi` needs to be on the next line and we need locally another comment character to avoid getting spaces into the definition—a weird problem :-)

```
\begingroup
\catcode`\%=12
\catcode`\^^A=14
\long\gdef\do@add@percent@to@temptokena#1{\@temptokena\expandafter{#1%^^A
```

Can't be on the same line as the `%` — see above.

```
  }}
\endgroup
```

(*End definition for* `\add@percent@to@temptokena`*. This function is documented on page* **??**.)

\@writefile

```
\long\def\@writefile#1#2{%
  \@ifundefined{tf@#1}\relax
    {%
```

If we write to the file we first prepare #2 using \add@percent@to@temptokena and then write the token register out.

```
      \add@percent@to@temptokena
        \@empty#2\protected@file@percent
        \add@percent@to@temptokena
     \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
    }%
}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\protected@file@percent}{Mask line endings}%
⟨latexrelease⟩\let\protected@file@percent\@undefined
⟨latexrelease⟩\let\add@percent@to@temptokena\@undefined
⟨latexrelease⟩\let\do@add@percent@to@temptokena\@undefined
⟨latexrelease⟩\let\dont@add@percent@to@temptokena\@undefined
⟨latexrelease⟩\long\def\@writefile#1#2{%
⟨latexrelease⟩  \@ifundefined{tf@#1}\relax
⟨latexrelease⟩    {\@temptokena{#2}%
⟨latexrelease⟩     \immediate\write\csname tf@#1\endcsname{\the\@temptokena}%
⟨latexrelease⟩    }%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@writefile*. This function is documented on page* ??.)

\stop

```
\def\stop{\clearpage\deadcycles\z@\let\par\@@par\@@end}
```

(*End definition for* \stop*. This function is documented on page* ??.)

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

```
\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
```

End of historical LaTeX 2.09 comments.

`\begin`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\begin}{Use hook system}%
\DeclareRobustCommand*\begin[1]{%
  \UseHook{env/#1/before}%
  \@ifundefined{#1}%
    {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
    {\def\reserved@a{\def\@currenvir{#1}%
        \edef\@currenvline{\on@line}%
        \@execute@begin@hook{#1}%
        \csname #1\endcsname}}%
  \@ignorefalse
  \begingroup\@endpefalse\reserved@a}
```

Before the `\document` code is executed we have to first undo the `\endgroup` as there should be none for this environment to avoid that changes on top-level unnecessarily go to TeX's savestack, and we have to initialize all hooks in the hook system. So we need to test for this environment name. But once it has be found all this testing is no longer needed and so we redefine `\@execute@begin@hook` to simply use the hook.

```
\def\@execute@begin@hook #1{%
```

```
  \expandafter\ifx\csname #1\endcsname\document
    \endgroup
    \gdef\@execute@begin@hook##1{\UseHook{env/##1/begin}}%
    \@expl@@@initialize@all@@
  \fi
```

If this is an environment before `\begin{document}` we just run the hook so this can be outside the test.

```
  \UseHook{env/#1/begin}%
}
```

The top level definition for `\end`. for an explanation see below (this is the same as the 2019 version where it was introduced, but for rollback we have to repeat it).

```
\edef\end
  {\unexpanded{%
     \romannumeral
       \ifx\protect\@typeset@protect
       \expandafter       %1
         \expandafter        %2
       \expandafter       %1
           \expandafter         %3 expands the \csname inside \end<space>
       \expandafter       %1
         \expandafter        %2  expands \end<space>
       \expandafter       %1     expands the \else
           \z@
       \else
         \expandafter\z@\expandafter\protect
       \fi
   }%
   \expandafter\noexpand\csname end \endcsname
  }
```

Version that adds hooks (so different from the 2019 version). It fixes tlb3722 but the change should perhaps be made in `tabularx` instead.

```
\@namedef{end }#1{%
  \romannumeral
    \IfHookEmptyTF{env/#1/end}%
        {\expandafter\z@}%
        {\z@\UseHook{env/#1/end}}%
    \csname end#1\endcsname\@checkend{#1}%
    \expandafter\endgroup\if@endpe\@doendpe\fi
    \UseHook{env/#1/after}%
    \if@ignore\@ignorefalse\ignorespaces\fi
}
```

Version without the fix for tlb3722 for the record:

```
%\@namedef{end }#1{%
%  \UseHook{env/#1/end}%
%  \csname end#1\endcsname\@checkend{#1}%
%  \expandafter\endgroup\if@endpe\@doendpe\fi
%  \UseHook{env/#1/after}%
%  \if@ignore\@ignorefalse\ignorespaces\fi}%
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\begin}{Making \begin/\end robust}%
⟨latexrelease⟩\DeclareRobustCommand\begin[1]{%
⟨latexrelease⟩  \@ifundefined{#1}%
⟨latexrelease⟩    {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
⟨latexrelease⟩    {\def\reserved@a{\def\@currenvir{#1}%
⟨latexrelease⟩     \edef\@currenvline{\on@line}%
⟨latexrelease⟩     \csname #1\endcsname}}%
⟨latexrelease⟩  \@ignorefalse
⟨latexrelease⟩  \begingroup\@endpefalse\reserved@a}
```

A version that doesn't start out with `\relax` when in typesetting mode would be the following, but since `\begin` issues a `\begingroup` it wouldn't help much with respect to allowing things like `\noalign` or `\multicolumn` inside.

```
%\edef\begin
%  {\unexpanded{%
%       \ifx\protect\@typeset@protect
%         \expandafter\@gobble
%       \fi
%       \protect
%     }%
%   \expandafter\noexpand\csname begin \endcsname
%  }
%\@namedef{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}
```

While `\begin` was made robust simply by using `\DeclareRobustCommand` we need to be a bit more subtle with `\end` as there are packages out there that try to look into the top-level contents of `\end{foo}` (that is at the expansion of `\endfoo`) to see if it contains certain macros. This is done by hitting `\end{foo}` with three `\expandafter`s, the first to get

```
\csname endfoo\endcsname       \@checkend{foo}% etc.
```

the second to expand the `\csname`, i.e., to get to

```
\endfoo                        \@checkend{foo}% etc.
```

and the third to finally get to the top-level content of `\endfoo`, i.e.

```
<top-level content of \endfoo> \@checkend{foo}% etc.
```

Therefore a robust replacement should produce the same results after three expansions (there first is obviously different).

Basically the definition of `\end` should either produce `\protect\end␣` (when not doing typesetting) or it should produce `\end␣` (without the `\protect`) when doing typesetting. Furthermore, it should (when in typesetting mode) show exactly the same result as `\end␣` (which is the original fragile definition of `\end`) when you expand either of them twice, i.e.,

```
\endfoo                          \@checkend{foo}% etc.
```

That is achieved with the code below (which is worth studying carefully).

There is some trickery involved here: in particular we use `\romannumeral` to change a single expansion into three successive expansions in one go. That primitive expands until it has scanned a number (0 in this case, so it doesn't produce any output) and so it allows us to place arbitrary many `\expandafter`s inside that are all going to be executed when `\romannumeral` is hit by a single `\expandafter`.

```
⟨latexrelease⟩\edef\end
⟨latexrelease⟩  {\unexpanded{%
⟨latexrelease⟩     \romannumeral
⟨latexrelease⟩       \ifx\protect\@typeset@protect
⟨latexrelease⟩       \expandafter       %1
⟨latexrelease⟩         \expandafter        %2
⟨latexrelease⟩       \expandafter       %1
⟨latexrelease⟩           \expandafter         %3 expands the \csname inside \end<space>
⟨latexrelease⟩       \expandafter       %1
⟨latexrelease⟩         \expandafter        %2  expands \end<space>
⟨latexrelease⟩       \expandafter       %1     expands the \else
⟨latexrelease⟩           \z@
⟨latexrelease⟩       \else
⟨latexrelease⟩         \expandafter\z@\expandafter\protect
⟨latexrelease⟩       \fi
⟨latexrelease⟩   }%
⟨latexrelease⟩   \expandafter\noexpand\csname end \endcsname
⟨latexrelease⟩  }
```

And here is the original definition of `\end` the way it was in LaTeX for several decades now hidden in `\end␣`.

```
⟨latexrelease⟩\@namedef{end }#1{%
⟨latexrelease⟩  \csname end#1\endcsname\@checkend{#1}%
⟨latexrelease⟩  \expandafter\endgroup\if@endpe\@doendpe\fi
⟨latexrelease⟩  \if@ignore\@ignorefalse\ignorespaces\fi}
⟨latexrelease⟩\EndIncludeInRelease
```

An here the rollback in case that is ever needed.

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\begin}{Making \begin/\end robust}%
⟨latexrelease⟩\def\begin#1{%
⟨latexrelease⟩  \@ifundefined{#1}%
⟨latexrelease⟩    {\def\reserved@a{\@latex@error{Environment #1 undefined}\@eha}}%
⟨latexrelease⟩    {\def\reserved@a{\def\@currenvir{#1}%
⟨latexrelease⟩     \edef\@currenvline{\on@line}%
⟨latexrelease⟩     \csname #1\endcsname}}%
⟨latexrelease⟩  \@ignorefalse
⟨latexrelease⟩  \begingroup\@endpefalse\reserved@a}
⟨latexrelease⟩\def\end#1{%
⟨latexrelease⟩  \csname end#1\endcsname\@checkend{#1}%
⟨latexrelease⟩  \expandafter\endgroup\if@endpe\@doendpe\fi
⟨latexrelease⟩  \if@ignore\@ignorefalse\ignorespaces\fi}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\begin` *and* `\end`*. These functions are documented on page* **??**.)

\@checkend

```
\def\@checkend#1{\def\reserved@a{#1}\ifx
      \reserved@a\@currenvir \else\@badend{#1}\fi}
```

(*End definition for* `\@checkend`*. This function is documented on page* **??**.)

\@currenvline

We do need a default value for `\@currenvline` on top-level since the document environment cancels the brace group. This means that a mismatch with `\begin{document}` will not produce a line number. Thus the outer default must be `\@empty` or we will end up with two spaces.

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

(*End definition for* `\@currenvline`*. This function is documented on page* **??**.)

\AtBeginEnvironment
\AtEndEnvironment
\BeforeBeginEnvironment
\AfterEndEnvironment

We provide 4 high-level hook interfaces directly, the others only when etoolbox is loaded

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\AtBeginEnvironment}{Hooks for environments}%
\newcommand\AtBeginEnvironment[2][.]     {\AddToHook{env/#2/begin}[#1]}
\newcommand\AtEndEnvironment[2][.]       {\AddToHook{env/#2/end}[#1]}
\newcommand\BeforeBeginEnvironment[2][.]{\AddToHook{env/#2/before}[#1]}
\newcommand\AfterEndEnvironment[2][.]    {\AddToHook{env/#2/after}[#1]}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\AtBeginEnvironment}{Hooks for environments}%
⟨latexrelease⟩
⟨latexrelease⟩\let\AtBeginEnvironment\@undefined
⟨latexrelease⟩\let\AtEndEnvironment\@undefined
⟨latexrelease⟩\let\BeforeBeginEnvironment\@undefined
⟨latexrelease⟩\let\AfterEndEnvironment\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\AtBeginEnvironment` *and others. These functions are documented on page* 116.)

1.2 Center, Flushright, Flushleft

```
\message{center,}
```

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

```
\center, \flushright and \flushleft set
  \rightskip = 0pt or \@flushglue (as appropriate)
  \leftskip  = 0pt or \@flushglue (as appropriate)
  \parindent = 0pt
  \parfillskip    = 0pt. (except \flushleft)
  \\          == \par \vskip -\parskip
  \\[LENGTH] == \\ \vskip LENGTH
  \\*         == \par \penalty 10000 \vskip -\parskip
```

*[LEN] == * \vskip LENGTH

They invoke the trivlist environment to handle vertical spacing before and after them.

\centering, \raggedright and \raggedleft are the declaration analogs of the above.

\raggedright has a more universal effect, however. It sets \@rightskip := flushglue. Every environment, like the list environments,
that set \rightskip to its 'normal' value set it to \@rightskip

End of historical LaTeX 2.09 comments.

\@centercr

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                  {\@centercr}{Make robust}%
\protected\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
        \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@centercr}{Make robust}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@centercr{\ifhmode \unskip\else \@nolnerr\fi
⟨latexrelease⟩        \par\@ifstar{\nobreak\@xcentercr}\@xcentercr}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@centercr*. This function is documented on page* ??.)

\@xcentercr

```
\def\@xcentercr{\addvspace{-\parskip}\@ifnextchar
    [\@icentercr\ignorespaces}
```

(*End definition for* \@xcentercr*. This function is documented on page* ??.)

\@icentercr

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@icentercr}{centering, etc support calc}%
\def\@icentercr[#1]{\@vspace@calcify{#1}\ignorespaces}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@icentercr}{centering, etc support calc}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@icentercr[#1]{\vskip #1\ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@icentercr`*. This function is documented on page* **??**.)

`center` We use `\relax` to prevent `\item` scanning too far.

```
\def\center{\trivlist \centering\item\relax}
\def\endcenter{\endtrivlist}
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\centering}{Set finaldhypendemerits}%
```

`\centering`

```
\DeclareRobustCommand\centering{%
  \let\\\@centercr
  \rightskip\@flushglue\leftskip\@flushglue
  \finalhyphendemerits=\z@
  \parindent\z@\parfillskip\z@skip}
```

(*End definition for* `\centering`*. This function is documented on page* **??**.)

`\raggedright`

```
\DeclareRobustCommand\raggedright{%
  \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
  \finalhyphendemerits=\z@
  \leftskip\z@skip
  \parindent\z@}
```

(*End definition for* `\raggedright`*. This function is documented on page* **??**.)

`\raggedleft`

```
\DeclareRobustCommand\raggedleft{%
  \let\\\@centercr
  \rightskip\z@skip\leftskip\@flushglue
  \finalhyphendemerits=\z@
  \parindent\z@\parfillskip\z@skip}
```

(*End definition for* `\raggedleft`*. This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\centering}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\centering{%
⟨latexrelease⟩   \let\\\@centercr
⟨latexrelease⟩   \rightskip\@flushglue\leftskip\@flushglue
⟨latexrelease⟩   \parindent\z@\parfillskip\z@skip}
⟨latexrelease⟩\DeclareRobustCommand\raggedright{%
```

```
⟨latexrelease⟩  \let\\\@centercr\@rightskip\@flushglue \rightskip\@rightskip
⟨latexrelease⟩  \leftskip\z@skip
⟨latexrelease⟩  \parindent\z@}
⟨latexrelease⟩\DeclareRobustCommand\raggedleft{%
⟨latexrelease⟩  \let\\\@centercr
⟨latexrelease⟩  \rightskip\z@skip\leftskip\@flushglue
⟨latexrelease⟩  \parindent\z@\parfillskip\z@skip}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\centering}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\centering
⟨latexrelease⟩\kernel@make@fragile\raggedright
⟨latexrelease⟩\kernel@make@fragile\raggedleft
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\@rightskip`

```
\newskip\@rightskip \@rightskip \z@skip
```

(*End definition for* `\@rightskip`*. This function is documented on page* **??**.)

`flushleft` We use `\relax` to prevent `\item` scanning too far.

```
\def\flushleft{\trivlist \raggedright\item\relax}
\def\endflushleft{\endtrivlist}
```

`flushright` We use `\relax` to prevent `\item` scanning too far.

```
\def\flushright{\trivlist \raggedleft\item\relax}
\def\endflushright{\endtrivlist}
```

1.3 Verbatim

```
\message{verbatim,}
```

The verbatim environment uses the fixed-width `\ttfamily` font, turns blanks into spaces, starts a new line for each carriage return (or sequence of consecutive carriage returns), and interprets *every* character literally. I.e., all special characters `\, {, $`, etc. are `\catcode`'d to 'other'.

The command `\verb` produces in-line verbatim text, where the argument is delimited by any pair of characters. E.g., `\verb #...#` takes '`...`' as its argument, and sets it verbatim in `\ttfamily` font.

The *-variants of these commands are the same, except that spaces print as the TEXbook's space character instead of as blank spaces.

`\@vobeyspaces`

```
{\catcode`\ =\active%
\gdef\@vobeyspaces{\catcode`\ \active\let \@xobeysp}}
```

(*End definition for* `\@vobeyspaces`*. This function is documented on page* **??**.)

`\@xobeysp`

(*End definition for* `\@xobeysp`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@xverbatim` *and* `\@sxverbatim`*. These functions are documented on page* **??**.)

`\@verbatim` Real start of verbatim environment We use `\relax` to prevent `\item` scanning too far.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2017-04-15}{\@verbatim}%
⟨latexrelease⟩                  {Disable hyphenation in verbatim}%
\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). Switch language when in vertical mode.

```
  \@@par
```

Set `\language` here to suppress hyphenation. Done this way rather than setting `\hyphenchar` as that is a global setting.

```
  \language\l@nohyphenation
  \@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
```

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}%
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\@verbatim}%
⟨latexrelease⟩                  {Disable hyphenation in verbatim}%
⟨latexrelease⟩\def\@verbatim{\trivlist \item\relax
⟨latexrelease⟩  \if@minipage\else\vskip\parskip\fi
⟨latexrelease⟩  \leftskip\@totalleftmargin\rightskip\z@skip
⟨latexrelease⟩  \parindent\z@\parfillskip\@flushglue\parskip\z@skip
```

```
⟨latexrelease⟩  \@@par
⟨latexrelease⟩  \@tempswafalse
⟨latexrelease⟩  \def\par{%
⟨latexrelease⟩    \if@tempswa
⟨latexrelease⟩      \leavevmode \null \@@par\penalty\interlinepenalty
⟨latexrelease⟩    \else
⟨latexrelease⟩      \@tempswatrue
⟨latexrelease⟩      \ifhmode\@@par\penalty\interlinepenalty\fi
⟨latexrelease⟩    \fi}%
⟨latexrelease⟩  \let\do\@makeother \dospecials
⟨latexrelease⟩  \obeylines \verbatim@font \@noligs
⟨latexrelease⟩  \hyphenchar\font\m@ne
⟨latexrelease⟩  \everypar \expandafter{\the\everypar \unpenalty}%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@verbatim`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\verbatim` *and* `\endverbatim`*. These functions are documented on page* **??**.)

`\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}
```

(*End definition for* `\verbatim@font`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\verbvisiblespace}{Setup visible space for \verb}%
```

`\asciispace`

The character in slot 32, in typewriter fonts (historically) a visible space but in other fonts a real space or something else

```
\DeclareRobustCommand\asciispace{\char 32 }
```

(*End definition for* `\asciispace`*. This function is documented on page* **??**.)

`\verbvisiblespace`

This defines how to get a visible space in `\verb*` and friends. In classic TEX this is just the slot 32, but in TU encoded fonts we switch fonts and take the character from cmtt.

```
\ifx\Umathcode\@undefined
  \let\verbvisiblespace\asciispace                                % Pdftex version
\else
  \DeclareRobustCommand\verbvisiblespace
           {\leavevmode{\usefont{OT1}{cmtt}{m}{n}\asciispace}}    % xetex/luatex version
\fi
```

(*End definition for* `\verbvisiblespace`*. This function is documented on page* **??**.)

`\@setupverbvisiblespace` In pdfTEX a catcode 12 space will produce the character in slot 32 which is assumed to be a visible space character (in a typewriter font in OT1 or T1 encoding). In XeTEX or LuaTEX a font in TU encoding is normally used and that has a real space in this slot. So what we do in this case is this: we check the definition of `\verbvisiblespace` and if it is `\asciispace` we assume that the char32 can be used (e.g., in pdfTEX). We then redefine `\@xobeysp` so that after running `\@vobeyspaces` we get characters from slot 32 for each active space.

```
\def\@setupverbvisiblespace{%
  \ifx\verbvisiblespace\asciispace
    \let\@xobeysp\asciispace
  \else
```

Otherwise we measure the width of a character in the mon-spaced current font and place a `\verbvisiblespace` into a box of the right width which we are then using as the character for a space. By default this will be the space character from OT1 cmtt but by changing `\verbvisiblespace` one could use, for example, the `\textvisiblespace` of the current typewriter font.

```
    \setbox\z@\hbox{x}%
    \setbox\@verbvisiblespacebox\hbox to\wd\z@{\hss\verbvisiblespace\hss}%
    \def\@xobeysp{\leavevmode\copy\@verbvisiblespacebox}%
  \fi
}
```

(*End definition for* `\@setupverbvisiblespace`*. This function is documented on page* **??**.)

`\@verbvisiblespacebox` The box to hold the visible space character if it isn't in slot 32 in the current typewriter font.

```
\newbox\@verbvisiblespacebox
```

(*End definition for* `\@verbvisiblespacebox`*. This function is documented on page* **??**.)

`verbatim*` For `verbatim*` we also set up the correct visible space character definition and then run `\@vobeyspaces`. As this code is not called as part of the normal verbatim environment (the method is done the other way around this time) we don't have to check if space is already active—it shouldn't be.

```
\@namedef{verbatim*}{\@verbatim
  \@setupverbvisiblespace
  \frenchspacing\@vobeyspaces\@sxverbatim}
\expandafter\let\csname endverbatim*\endcsname =\endverbatim
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\verbvisiblespace}{Setup visible space for \verb}%
⟨latexrelease⟩
⟨latexrelease⟩\@namedef{verbatim*}{\@verbatim\@sxverbatim}
⟨latexrelease⟩
⟨latexrelease⟩\let\asciispace            \@undefined
⟨latexrelease⟩\let\verbvisiblespace      \@undefined
⟨latexrelease⟩\let\@setupverbvisiblespace\@undefined
⟨latexrelease⟩\let\@verbvisiblespacebox  \@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

\@sverb
\@@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.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@sverb}{Drop spaces before \verb delimiter}%
```

If the users types \verb !~! foo then surprisingly we would get the space as the delimiter and thus "!~!foo" in the output. To avoid this scenario we check if #1 has the character code of a space, if so we recurse otherwise we call \@@sverb (which is the original definition of \@sverb.

```
\def\@sverb#1{\if\noexpand#1 \expandafter\@sverb\else\@@sverb{#1}\fi}
```

```
\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}%
```

If \@sverb is called from \@verb then space is already active and supposed to produce a real space. In this case we do nothing. Otherwise we run \@setupverbvisiblespace to setup the right visible space char and afterwards \@vobeyspaces to make it the definition for the active space character.

```
  \ifnum\catcode`\ =\active
  \else  \@setupverbvisiblespace \@vobeyspaces \fi
}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\@sverb}{Setup visible space for \verb}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@sverb#1{%
⟨latexrelease⟩  \catcode`#1\active
⟨latexrelease⟩  \lccode`\~`#1%
⟨latexrelease⟩  \gdef\verb@balance@group{\verb@egroup
⟨latexrelease⟩     \@latex@error{\noexpand\verb illegal in command argument}\@ehc}%
⟨latexrelease⟩  \aftergroup\verb@balance@group
⟨latexrelease⟩  \lowercase{\let~\verb@egroup}%
⟨latexrelease⟩  \ifnum\catcode`\ =\active
⟨latexrelease⟩  \else  \@setupverbvisiblespace \@vobeyspaces \fi
⟨latexrelease⟩}
⟨latexrelease⟩\let\@@sverb\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@sverb}{Setup visible space for \verb}%
⟨latexrelease⟩\def\@sverb#1{%
⟨latexrelease⟩  \catcode`#1\active
⟨latexrelease⟩  \lccode`\~`#1%
⟨latexrelease⟩  \gdef\verb@balance@group{\verb@egroup
⟨latexrelease⟩     \@latex@error{\noexpand\verb illegal in command argument}\@ehc}%
```

```
⟨latexrelease⟩   \aftergroup\verb@balance@group
⟨latexrelease⟩   \lowercase{\let~\verb@egroup}}%
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@sverb` *and* `\@@sverb`*. These functions are documented on page* **??**.)

`\@makeother`

```
\def\@makeother#1{\catcode`#112\relax}
```

(*End definition for* `\@makeother`*. This function is documented on page* **??**.)

`\verb@balance@group`

```
\let\verb@balance@group\@empty
```

(*End definition for* `\verb@balance@group`*. This function is documented on page* **??**.)

`\verb@egroup`

```
\def\verb@egroup{\global\let\verb@balance@group\@empty\egroup}
```

(*End definition for* `\verb@egroup`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\verb@eol@error`*. This function is documented on page* **??**.)

`\verb` Typesetting a small piece verbatim.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2017-04-15}{\verb}%
⟨latexrelease⟩                 {Disable hyphenation in verb}%
\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
  \bgroup
    \verb@eol@error \let\do\@makeother \dospecials
    \verbatim@font\@noligs
```

Set `\language` here to suppress hyphenation. Done this way rather than setting `\hyphenchar` as that is a global setting.

```
    \language\l@nohyphenation
    \@ifstar\@sverb\@verb}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}{\verb}%
⟨latexrelease⟩                 {Disable hyphenation in verb}%
⟨latexrelease⟩\def\verb{\relax\ifmmode\hbox\else\leavevmode\null\fi
⟨latexrelease⟩  \bgroup
⟨latexrelease⟩    \verb@eol@error \let\do\@makeother \dospecials
⟨latexrelease⟩    \verbatim@font\@noligs
⟨latexrelease⟩    \@ifstar\@sverb\@verb}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\verb`*. This function is documented on page* **??**.)

`\@verb`

```
\def\@verb{\@vobeyspaces \frenchspacing \@sverb}
```

(*End definition for* `\@verb`*. This function is documented on page* **??**.)

`\verbatim@nolig@list`

```
\def\verbatim@nolig@list{\do\`\do\<\do\>\do\,\do\'\do\-}
```

(*End definition for* `\verbatim@nolig@list`*. This function is documented on page* **??**.)

`\do@noligs`

```
\def\do@noligs#1{%
  \catcode`#1\active
  \begingroup
     \lccode`\~`#1\relax
     \lowercase{\endgroup\def~{\leavevmode\kern\z@\char`#1}}}
```

(*End definition for* `\do@noligs`*. This function is documented on page* **??**.)

`\@noligs` To stay compatible with packages that use `\@noligs` we keep it.

```
\def\@noligs{\let\do\do@noligs \verbatim@nolig@list}
```

(*End definition for* `\@noligs`*. This function is documented on page* **??**.)

⟨/2ekernel⟩

File E
ltmath.dtx

1 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,}
```

1.1 Math commands based on plain TEX

1.1.1 The log-like functions

\log The standard operators:

```
\DeclareRobustCommand\log{\mathop{\operator@font log}\nolimits}
\DeclareRobustCommand\lg{\mathop{\operator@font lg}\nolimits}
\DeclareRobustCommand\ln{\mathop{\operator@font ln}\nolimits}
\DeclareRobustCommand\lim{\mathop{\operator@font lim}}
\DeclareRobustCommand\limsup{\mathop{\operator@font lim\,sup}}
\DeclareRobustCommand\liminf{\mathop{\operator@font lim\,inf}}
\DeclareRobustCommand\sin{\mathop{\operator@font sin}\nolimits}
\DeclareRobustCommand\arcsin{\mathop{\operator@font arcsin}\nolimits}
\DeclareRobustCommand\sinh{\mathop{\operator@font sinh}\nolimits}
\DeclareRobustCommand\cos{\mathop{\operator@font cos}\nolimits}
\DeclareRobustCommand\arccos{\mathop{\operator@font arccos}\nolimits}
\DeclareRobustCommand\cosh{\mathop{\operator@font cosh}\nolimits}
\DeclareRobustCommand\tan{\mathop{\operator@font tan}\nolimits}
\DeclareRobustCommand\arctan{\mathop{\operator@font arctan}\nolimits}
\DeclareRobustCommand\tanh{\mathop{\operator@font tanh}\nolimits}
\DeclareRobustCommand\cot{\mathop{\operator@font cot}\nolimits}
\DeclareRobustCommand\coth{\mathop{\operator@font coth}\nolimits}
\DeclareRobustCommand\sec{\mathop{\operator@font sec}\nolimits}
\DeclareRobustCommand\csc{\mathop{\operator@font csc}\nolimits}
\DeclareRobustCommand\max{\mathop{\operator@font max}}
\DeclareRobustCommand\min{\mathop{\operator@font min}}
\DeclareRobustCommand\sup{\mathop{\operator@font sup}}
\DeclareRobustCommand\inf{\mathop{\operator@font inf}}
\DeclareRobustCommand\arg{\mathop{\operator@font arg}\nolimits}
\DeclareRobustCommand\ker{\mathop{\operator@font ker}\nolimits}
\DeclareRobustCommand\dim{\mathop{\operator@font dim}\nolimits}
\DeclareRobustCommand\hom{\mathop{\operator@font hom}\nolimits}
\DeclareRobustCommand\det{\mathop{\operator@font det}}
\DeclareRobustCommand\exp{\mathop{\operator@font exp}\nolimits}
\DeclareRobustCommand\Pr{\mathop{\operator@font Pr}}
\DeclareRobustCommand\gcd{\mathop{\operator@font gcd}}
\DeclareRobustCommand\deg{\mathop{\operator@font deg}\nolimits}
```

(*End definition for* `\log`*. This function is documented on page* **??**.)

\bmod And some operators have to be done by hand:

```
\DeclareRobustCommand\bmod{%
  \nonscript\mskip-\medmuskip\mkern5mu%
  \mathbin{\operator@font mod}\penalty900\mkern5mu%
  \nonscript\mskip-\medmuskip}
```

(*End definition for* `\bmod`*. This function is documented on page* **??**.)

`\pmod`

```
\DeclareRobustCommand\pmod[1]{%
  \allowbreak\mkern18mu({\operator@font mod}\,\,#1)}
```

(*End definition for* `\pmod`*. This function is documented on page* **??**.)

1.1.2 Biggggg

`\big` Variants on `\big` and friends for use with delimiters:

```
\DeclareRobustCommand\bigl{\mathopen\big}
\DeclareRobustCommand\bigm{\mathrel\big}
\DeclareRobustCommand\bigr{\mathclose\big}
\DeclareRobustCommand\Bigl{\mathopen\Big}
\DeclareRobustCommand\Bigm{\mathrel\Big}
\DeclareRobustCommand\Bigr{\mathclose\Big}
\DeclareRobustCommand\biggl{\mathopen\bigg}
\DeclareRobustCommand\biggm{\mathrel\bigg}
\DeclareRobustCommand\biggr{\mathclose\bigg}
\DeclareRobustCommand\Biggl{\mathopen\Bigg}
\DeclareRobustCommand\Biggm{\mathrel\Bigg}
\DeclareRobustCommand\Biggr{\mathclose\Bigg}
```

(*End definition for* `\big`*. This function is documented on page* **??**.)

1.1.3 The UNSORTED Rest

The other math commands are lifted from plain TEX.

`\jot`

```
\newdimen\jot
\jot=3pt
```

(*End definition for* `\jot`*. This function is documented on page* **??**.)

`\interdisplaylinepenalty`

```
\newcount\interdisplaylinepenalty
\interdisplaylinepenalty=100
```

(*End definition for* `\interdisplaylinepenalty`*. This function is documented on page* **??**.)

`\choose`

```
\def\choose{\atopwithdelims()}
```

(*End definition for* `\choose`*. This function is documented on page* **??**.)

`\brack`

```
\def\brack{\atopwithdelims[]}
```

(*End definition for* `\brack`*. This function is documented on page* **??**.)

\brace

```
\def\brace{\atopwithdelims\{\}}
```

(*End definition for* \brace*. This function is documented on page* ??.)

\mathpalette

```
\def\mathpalette#1#2{%
  \mathchoice
    {#1\displaystyle{#2}}%
    {#1\textstyle{#2}}%
    {#1\scriptstyle{#2}}%
    {#1\scriptscriptstyle{#2}}}
```

(*End definition for* \mathpalette*. This function is documented on page* ??.)

\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@}
```

(*End definition for* \root*,* \rootbox*, and* \r@@t*. These functions are documented on page* ??.)

\phantom
\hphantom
\vphantom

```
\newif\ifv@
\newif\ifh@
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\vphantom}{Make commands robust}%
\DeclareRobustCommand\vphantom{\v@true\h@false\ph@nt}
\DeclareRobustCommand\hphantom{\v@false\h@true\ph@nt}
\DeclareRobustCommand\phantom{\v@true\h@true\ph@nt}
```

\mathstrut

```
\DeclareRobustCommand\mathstrut{\vphantom(}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\vphantom}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\vphantom
⟨latexrelease⟩\kernel@make@fragile\hphantom
⟨latexrelease⟩\kernel@make@fragile\phantom
⟨latexrelease⟩\kernel@make@fragile\mathstrut
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\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}
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\finph@nt}{Start LR-mode}%
\def\finph@nt{%
  \setbox\tw@\null
  \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
  \ifh@ \wd\tw@\wd\z@\fi

  \leavevmode@ifvmode\box\tw@}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\finph@nt}{Start LR-mode}%
⟨latexrelease⟩\def\finph@nt{%
⟨latexrelease⟩  \setbox\tw@\null
⟨latexrelease⟩  \ifv@ \ht\tw@\ht\z@ \dp\tw@\dp\z@\fi
⟨latexrelease⟩  \ifh@ \wd\tw@\wd\z@\fi \box\tw@}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\phantom` *and others. These functions are documented on page* **??**.)

`\smash`

```
\DeclareRobustCommand\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}

⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\finsm@sh}{Start LR-mode}%
\def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \leavevmode@ifvmode\box\z@}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\finsm@sh}{Start LR-mode}%
⟨latexrelease⟩\def\finsm@sh{\ht\z@\z@ \dp\z@\z@ \box\z@}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\smash`*. This function is documented on page* ??.)

`\buildrel`

```
\def\buildrel#1\over#2{\mathrel{\mathop{\kern\z@#2}\limits^{#1}}}
```

(*End definition for* `\buildrel`*. This function is documented on page* ??.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\cases}{Make commands robust}%
```

`\cases`

```
\DeclareRobustCommand*\cases[1]{\left\{\,\vcenter{\normalbaselines\m@th
    \ialign{$##\hfil$&\quad{##}\hfil\crcr#1\crcr}}\right.}
```

(*End definition for* `\cases`*. This function is documented on page* ??.)

`\matrix`

```
\DeclareRobustCommand*\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}}}\,}
```

(*End definition for* `\matrix`*. This function is documented on page* ??.)

`\pmatrix`

```
\DeclareRobustCommand*\pmatrix[1]{\left(\matrix{#1}\right)}
```

(*End definition for* `\pmatrix`*. This function is documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\cases}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\cases
⟨latexrelease⟩\kernel@make@fragile\matrix
⟨latexrelease⟩\kernel@make@fragile\pmatrix
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\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}
```

(*End definition for* `\bordermatrix`*. This function is documented on page* **??**.)

`\openup`

```
\def\openup{\afterassignment\@penup\dimen@}
```

```
\def\@penup{\advance\lineskip\dimen@
  \advance\baselineskip\dimen@
  \advance\lineskiplimit\dimen@}
```

(*End definition for* `\openup`*. This function is documented on page* **??**.)

`\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}}
```

(*End definition for* `\displaylines`*. This function is documented on page* **??**.)

`\sp`
`\sb`

```
\let\sp=^
\let\sb=_
```

(*End definition for* `\sp` *and* `\sb`*. These functions are documented on page* **??**.)

`\tmspace`
`\ `
`\thinspace`
`\!`
`\negthinspace`
`\:`
`\medspace`
`\negmedspace`
`\;`
`\thickspace`
`\negthickspace`

Originally LaTeX only provided a small set of spacing commands for use in text and math, some of the commands like `\;` were only supported in manth mode. `amsmath` normalized and provided all of them in text and math. This code has now been moved to the kernel so that it is generally available.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\tmspace}{amsmath spacing commands}%
```

`\tmspace` is really meant to be an internal command so it doesn't necessarily has to be robust but it was robust in `amsmath` so we leave it like that.

```
\DeclareRobustCommand\tmspace[3]{%
  \ifmmode\mskip#1#2\else\leavevmode@ifvmode\kern#1#3\fi\relax}
```

In amsmath the text kern is .1667em. For compatibility reasons we keep the longer one.

```
\DeclareRobustCommand\,{\tmspace+\thinmuskip{.16667em}}
\let\thinspace\,
\DeclareRobustCommand\!{\tmspace-\thinmuskip{.16667em}}
\let\negthinspace\!
\DeclareRobustCommand\:{\tmspace+\medmuskip{.2222em}}
\let\medspace\:
```

LaTeX has a second name for this in its manual:

```
\let\>=\:
\DeclareRobustCommand\negmedspace{\tmspace-\medmuskip{.2222em}}
\DeclareRobustCommand\;{\tmspace+\thickmuskip{.2777em}}
\let\thickspace\;
\DeclareRobustCommand\negthickspace{\tmspace-\thickmuskip{.2777em}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\tmspace}{amsmath spacing commands}%
⟨latexrelease⟩
⟨latexrelease⟩\let\tmspace\@undefined
⟨latexrelease⟩\DeclareRobustCommand{\,}{%
⟨latexrelease⟩   \relax\ifmmode\mskip\thinmuskip\else\thinspace\fi}
⟨latexrelease⟩\DeclareRobustCommand\thinspace{\leavevmode@ifvmode\kern .16667em }
⟨latexrelease⟩\DeclareRobustCommand\negthinspace{\leavevmode@ifvmode\kern-.16667em }
⟨latexrelease⟩\def\>{\mskip\medmuskip}
⟨latexrelease⟩\let\:=\>
⟨latexrelease⟩\def\;{\mskip\thickmuskip}
⟨latexrelease⟩\def\!{\mskip-\thinmuskip}
⟨latexrelease⟩

⟨latexrelease⟩\let\negmedspace\@undefined
⟨latexrelease⟩\let\negthickspace\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \tmspace *and others. These functions are documented on page* ??.)

*

```
\DeclareRobustCommand\*{\discretionary{\thinspace\the\textfont2\char2}{}{}}
```

(*End definition for* *. *This function is documented on page* ??.)

\: Nickname for the medium space since \> is not available inside tabbing.

```
%\let\:=\>
```

(*End definition for* \:. *This function is documented on page* ??.)

\active@math@prime This is the definition of the active math prime.

```
\def\active@math@prime{^\bgroup\prim@s}
```

(*End definition for* \active@math@prime. *This function is documented on page* ??.)

\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}
```

(*End definition for* `\prime@s`*. This function is documented on page* **??**.)

```
{\catcode`\_=\active \gdef_{\_}} % _ in math is
                                 % either subscript or \_
```

1.2 Math Environments

`\(` `\)` Produces `$...$` with checks that `\(` isn't used in math mode, and that `\)` is only used in math mode begun with `\(`.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\(}{Make \( robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\({%
  \relax\ifmmode\@badmath\else$\fi}%
\DeclareRobustCommand\){%
  \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\(}{Make \( robust}%
⟨latexrelease⟩\def\({%
⟨latexrelease⟩  \relax\ifmmode\@badmath\else$\fi}%
⟨latexrelease⟩\expandafter\let\csname\string( \endcsname\@undefined
⟨latexrelease⟩\def\){%
⟨latexrelease⟩  \relax\ifmmode\ifinner$\else\@badmath\fi\else \@badmath\fi}%
⟨latexrelease⟩\expandafter\let\csname\string) \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\(` *and* `\)`*. These functions are documented on page* **??**.)

`\[` `\]` Produces `$$...$$` with checks that `\[` isn't used in math mode, and that `\]` is only used in display math mode (though there is no real test that this display math started with `\[` and not with `$$`).

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\[}{Make \[ robust}%
```

```
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\[{%
   \relax\ifmmode
      \@badmath
   \else
      \ifvmode
         \nointerlineskip
         \makebox[.6\linewidth]{}%
      \fi
      $$%%$$ BRACE MATCH HACK
   \fi
}%
\DeclareRobustCommand\]{%
   \relax\ifmmode
      \ifinner
         \@badmath
      \else
         $$%%$$ BRACE MATCH HACK
      \fi
   \else
      \@badmath
   \fi
   \ignorespaces
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\[}{Make \[ robust}%
⟨latexrelease⟩\def\[{%
⟨latexrelease⟩   \relax\ifmmode
⟨latexrelease⟩      \@badmath
⟨latexrelease⟩   \else
⟨latexrelease⟩      \ifvmode
⟨latexrelease⟩         \nointerlineskip
⟨latexrelease⟩         \makebox[.6\linewidth]{}%
⟨latexrelease⟩      \fi
⟨latexrelease⟩      $$%%$$ BRACE MATCH HACK
⟨latexrelease⟩   \fi
⟨latexrelease⟩}%
⟨latexrelease⟩\expandafter\let\csname\string[ \endcsname\@undefined
⟨latexrelease⟩\def\]{%
⟨latexrelease⟩   \relax\ifmmode
⟨latexrelease⟩      \ifinner
⟨latexrelease⟩         \@badmath
⟨latexrelease⟩      \else
⟨latexrelease⟩         $$%%$$ BRACE MATCH HACK
⟨latexrelease⟩      \fi
⟨latexrelease⟩   \else
⟨latexrelease⟩      \@badmath
⟨latexrelease⟩   \fi
⟨latexrelease⟩   \ignorespaces
⟨latexrelease⟩}%
⟨latexrelease⟩\expandafter\let\csname\string] \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨*2ekernel⟩
```

(*End definition for* `\[` *and* `\]`*. These functions are documented on page* **??**.)

`math`
`displaymath`

Disguises for `\(...\)` and `\[...\]`.

```
\let\math=\(
\let\endmath=\)
\def\displaymath{\[}
\def\enddisplaymath{\]\@ignoretrue}
```

`\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}
```

(*End definition for* `\c@equation`*. This function is documented on page* **??**.)

`\@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)}}
```

(*End definition for* `\@eqnnum`*. This function is documented on page* **??**.)

`\stackrel`

A disguise for plain TEX's buildrel.

```
\DeclareRobustCommand\stackrel[2]{\mathrel{\mathop{#2}\limits^{#1}}}
```

(*End definition for* `\stackrel`*. This function is documented on page* **??**.)

`\frac`

A disguise for plain TEX's `\over`.

```
\DeclareRobustCommand\frac[2]{{\begingroup#1\endgroup\over#2}}
```

(*End definition for* `\frac`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\sqrt` *and* `\@sqrt`*. These functions are documented on page* **??**.)

`eqnarray`
`\@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
```

(*End definition for* \@eqcnt *and others. These functions are documented on page* ??.)

\nonumber Switches off equation numbering.

```
\def\nonumber{\global\@eqnswfalse}
```

(*End definition for* \nonumber*. This function is documented on page* ??.)

\@eqncr
\@xeqncr
\@yeqncr

```
\def\@eqncr{%
   {\ifnum0=`}\fi
   \@ifstar{%
      \global\@eqpen\@M\@yeqncr
   }{%
      \global\@eqpen\interdisplaylinepenalty \@yeqncr
   }%
}
```

```
\def\@yeqncr{\@testopt\@xeqncr\z@skip}
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\@xeqncr}{eqnarray support calc syntax}%
\def\@xeqncr[#1]{%
   \ifnum0=`{\fi}%
   \@@eqncr
```

```
   \noalign{\penalty\@eqpen\vskip\jot\@vspace@calcify{#1}}%
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@xeqncr}{eqnarray support calc syntax}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@xeqncr[#1]{%
⟨latexrelease⟩   \ifnum0=`{\fi}%
⟨latexrelease⟩   \@@eqncr
⟨latexrelease⟩   \noalign{\penalty\@eqpen\vskip\jot\vskip #1\relax}%
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@eqncr`, `\@xeqncr`, *and* `\@yeqncr`. *These functions are documented on page* ??.)

`\@@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}
```

(*End definition for* `\@@eqncr`. *This function is documented on page* ??.)

`eqnarray*` `\@eqnarray*` Here's the eqnarray* environment:

```
\let\@seqncr=\@eqncr
\@namedef{eqnarray*}{\def\@eqncr{\nonumber\@seqncr}\eqnarray}
\@namedef{endeqnarray*}{\nonumber\endeqnarray}
```

(*End definition for* `\@seqncr`. *This function is documented on page* ??.)

`\lefteqn` `\lefteqn{FORMULA}` typesets `FORMULA` in display math style flushleft in a box of width zero.

```
\def\lefteqn#1{\rlap{$\displaystyle #1$}}
```

(*End definition for* `\lefteqn`. *This function is documented on page* ??.)

`\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}
```

(*End definition for* `\ensuremath`. *This function is documented on page* ??.)

\@ensuredmath The \relax stops \ensuremath{} starting display math.

```
\long\def\@ensuredmath#1{$\relax#1$}
```

(*End definition for* \@ensuredmath. *This function is documented on page* ??.)

```
⟨/2ekernel⟩
```

1.3 External options to the standard document classes

1.3.1 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⟩
```

(*End definition for* \@eqnnum. *This function is documented on page* ??.)

1.3.2 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⟩
\newskip\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}
```

(*End definition for* \mathindent. *This function is documented on page* ??.)

\[Begin display math;

```
\IncludeInRelease{2015/01/01}{\[}{Make \[ robust}%
\DeclareRobustCommand\[{\relax
                \ifmmode\@badmath
                \else
                  \begin{trivlist}%
                    \@beginparpenalty\predisplaypenalty
                    \@endparpenalty\postdisplaypenalty
                    \item[]\leavevmode
                    \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
                      \hskip\mathindent\bgroup
                \fi}
\EndIncludeInRelease
```

```
\IncludeInRelease{0000/00/00}{\[}{Make \[ robust}%
\renewcommand\[{\relax
                \ifmmode\@badmath
                \else
                  \begin{trivlist}%
                    \@beginparpenalty\predisplaypenalty
                    \@endparpenalty\postdisplaypenalty
                    \item[]\leavevmode
                    \hb@xt@\linewidth\bgroup $\m@th\displaystyle %$
                      \hskip\mathindent\bgroup
                \fi}
\EndIncludeInRelease
```

(*End definition for* `\[`*. This function is documented on page* ??.)

`\]` end display math;

```
\IncludeInRelease{2015/01/01}{\]}{Make \] robust}%
\DeclareRobustCommand\]{\relax
                \ifmmode
                      \egroup $\hfil% $
                    \egroup
                  \end{trivlist}%
                \else \@badmath
                \fi}
\EndIncludeInRelease
```

```
\IncludeInRelease{0000/00/00}{\]}{Make \] robust}%
\renewcommand\]{\relax
                \ifmmode
                      \egroup $\hfil% $
                    \egroup
                  \end{trivlist}%
                \else \@badmath
                \fi}
\EndIncludeInRelease
```

(*End definition for* `\]`*. This function is documented on page* ??.)

`equation` The equation environment

```
\renewenvironment{equation}%
    {\@beginparpenalty\predisplaypenalty
     \@endparpenalty\postdisplaypenalty
     \refstepcounter{equation}%
     \trivlist \item[]\leavevmode
       \hb@xt@\linewidth\bgroup $\m@th% $
         \displaystyle
         \hskip\mathindent}%
```

Ensure that there is at least a space between formula and equation number so that they don't bump in each other.

```
        {$\hskip .3em minus.3em\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 F
ltlists.dtx

1 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.

1.1 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.

1.2 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.

1.3 Penalties

\@beginparpenalty: put at the beginning of a list

\@endparpenalty: put at end of list

\@itempenalty: put between items.

1.4 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.

1.5 Default Values

Defaults for the list environment are set as follows. First, \rightmargin, \listparindent and \itemindent are set to 0pt. Then, one of the commands \@listi, \@listii, ... , \@listvi is called, depending upon the current level of the list. The \@list ... commands should be defined by the document style. A convention that the document style should follow is to set \leftmargin to \leftmargini,..., \leftmarginvi for the appropriate level. Items that aren't changed may be left alone, but everything that could possibly be changed must be reset. *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
\list{LABEL}{COMMANDS} ==
 BEGIN
   if \@listdepth > 5
     then LaTeX error: 'Too deeply nested'
     else \@listdepth :=G \@listdepth + 1
```

```
    fi
    \rightmargin      := 0pt
    \listparindent    := 0pt
    \itemindent       := 0pt
    \eval(@list \romannumeral\the\@listdepth)  %% Set default values:
    \@itemlabel       :=L LABEL
    \makelabel        == \@mklab
    @nmbrlist         :=L false
    COMMANDS

    \@trivlist                % commands common to \list and \trivlist

    \parskip          :=L \parsep
    \parindent        :=L \listparindent
    \linewidth        :=L \linewidth - \rightmargin -\leftmargin
    \@totalleftmargin :=L \@totalleftmargin + \leftmargin
    \parshape 1 \@totalleftmargin \linewidth
    \ignorespaces                     % gobble space up to \item
   END

\endlist == BEGIN \@listdepth :=G \@listdepth -1
                  \endtrivlist
            END

\@trivlist ==
 BEGIN
    if @newlist = T then \@noitemerr fi
                    %% This command removed for some forgotten reason.
    \@topsepadd :=L \topsep
    if @noskipsec then leave vertical mode fi  %% Added 11 Jun 85
    if vertical mode
      then \@topsepadd :=L \@topsepadd + \partopsep
      else \unskip \par             % remove glue from end of last line
    fi
    if @inlabel = true
       then @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`

End of historical LaTeX 2.09 comments.

`\topskip`
`\partopsep`
`\itemsep`
`\parsep`
`\@topsep`
`\@topsepadd`
`\outerparskip`

```
⟨*2ekernel⟩
\newskip\topsep
\newskip\partopsep
\newskip\itemsep
\newskip\parsep
\newskip\@topsep
\newskip\@topsepadd
\newskip\@outerparskip
```

(*End definition for* `\topskip` *and others. These functions are documented on page* **??**.)

`\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@
```

(*End definition for* `\leftmargin` *and others. These functions are documented on page* **??**.)

\leftmargini
\leftmarginii
\leftmarginiii
\leftmarginiv
\leftmarginv
\leftmarginvi

```
\newdimen\leftmargini
\newdimen\leftmarginii
\newdimen\leftmarginiii
\newdimen\leftmarginiv
\newdimen\leftmarginv
\newdimen\leftmarginvi
```

(*End definition for* `\leftmargini` *and others. These functions are documented on page* **??**.)

\@listdepth
\@itempenalty
\@beginparpenalty
\@endparpenalty

```
\newcount\@listdepth \@listdepth=0
\newcount\@itempenalty
\newcount\@beginparpenalty
\newcount\@endparpenalty
```

(*End definition for* `\@listdepth` *and others. These functions are documented on page* **??**.)

\@labels

```
\newbox\@labels
```

(*End definition for* `\@labels`*. This function is documented on page* **??**.)

\if@inlabel
\@inlabelfalse
\@inlabeltrue

```
\newif\if@inlabel \@inlabelfalse
```

(*End definition for* `\if@inlabel`*,* `\@inlabelfalse`*, and* `\@inlabeltrue`*. These functions are documented on page* **??**.)

\if@newlist
\@newlistfalse
\@newlisttrue

```
\newif\if@newlist   \@newlistfalse
```

(*End definition for* `\if@newlist`*,* `\@newlistfalse`*, and* `\@newlisttrue`*. These functions are documented on page* **??**.)

\if@noparitem
\@noparitemfalse
\@noparitemtrue

```
\newif\if@noparitem \@noparitemfalse
```

(*End definition for* `\if@noparitem`*,* `\@noparitemfalse`*, and* `\@noparitemtrue`*. These functions are documented on page* **??**.)

\if@noparlist
\@noparlistfalse
\@noparlisttrue

```
\newif\if@noparlist \@noparlistfalse
```

(*End definition for* `\if@noparlist`*,* `\@noparlistfalse`*, and* `\@noparlisttrue`*. These functions are documented on page* **??**.)

\if@noitemarg
\@noitemargfalse
\@noitemargtrue

```
\newif\if@noitemarg \@noitemargfalse
```

(*End definition for* `\if@noitemarg`*,* `\@noitemargfalse`*, and* `\@noitemargtrue`*. These functions are documented on page* **??**.)

\if@newlist
\@newlistfalse
\@newlisttrue

```
\newif\if@nmbrlist  \@nmbrlistfalse
```

(*End definition for* `\if@newlist`, `\@newlistfalse`, *and* `\@newlisttrue`. *These functions are documented on page* **??**.)

\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}
```

(*End definition for* `\list`. *This function is documented on page* **??**.)

`\par@deathcycles`

```
\newcount\par@deathcycles
```

(*End definition for* `\par@deathcycles`. *This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@trivlist`*. This function is documented on page* **??**.)

\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}}
```

(*End definition for* `\trivlist`*. This function is documented on page* **??**.)

\endlist

```
\def\endlist{%
  \global\advance\@listdepth\m@ne
  \endtrivlist}
```

(*End definition for* `\endlist`*. This function is documented on page* **??**.)

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
}
```

(*End definition for* \endtrivlist. *This function is documented on page* ??.)

\@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 \everypar, \@endparenv was changed to set the @endpe switch, letting \end redefine \par and \everypar.

This allows paragraph-making environments to work right when called by other environments. (Changed 27 Oct 86)

```
\def\@endparenv{%
  \addpenalty\@endparpenalty\addvspace\@topsepadd\@endpetrue}
```

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@doendpe}{clubpenalty fix}%
\def\@doendpe{\@endpetrue
     \def\par{\@restorepar
```

If a section heading changes \clubpenalty to keep lines after it together then this modification is restored via the \everypar mechanism at the start of the next paragraph. As we destroy the contents of this token here we explicity set \clubpenalty back to its default.

```
              \clubpenalty\@clubpenalty
              \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}}
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@doendpe}{clubpenalty fix}%
⟨latexrelease⟩\def\@doendpe{\@endpetrue
⟨latexrelease⟩     \def\par{\@restorepar\everypar{}\par\@endpefalse}\everypar
⟨latexrelease⟩                {{\setbox\z@\lastbox}\everypar{}\@endpefalse}}
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* \@endparenv *and* \@doendpe. *These functions are documented on page* ??.)

\if@endpe
\@endpefalse
\@endpeltrue

```
\newif\if@endpe
\@endpefalse
```

(*End definition for* `\if@endpe`, `\@endpefalse`, *and* `\@endpeltrue`. *These functions are documented on page* **??**.)

`\@mklab`

```
\def\@mklab#1{\hfil #1}
```

(*End definition for* `\@mklab`. *This function is documented on page* **??**.)

`\item`

```
\def\item{%
  \@inmatherr\item
  \@ifnextchar [\@item{\@noitemargtrue \@item[\@itemlabel]}}
```

(*End definition for* `\item`. *This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@donoparitem`. *This function is documented on page* **??**.)

`\@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.

```
    \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}
```

(*End definition for* `\@item`*. This function is documented on page* ??.)

`\makelabel`

```
\def\makelabel#1{%
  \@latex@error{Lonely \string\item--perhaps a missing
        list environment}\@ehc}
```

(*End definition for* `\makelabel`*. This function is documented on page* ??.)

`\@nbitem`

```
\def\@nbitem{%
  \@tempskipa\@outerparskip
  \advance\@tempskipa -\parskip
  \addvspace\@tempskipa}
```

(*End definition for* `\@nbitem`*. This function is documented on page* ??.)

`\usecounter`

```
\def\usecounter#1{\@nmbrlisttrue\def\@listctr{#1}\setcounter{#1}\z@}
```

(*End definition for* `\usecounter`*. This function is documented on page* ??.)

1.6 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`.

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

`\enumerate` ==
BEGIN

```
if \@enumdepth > 3
  then errormessage: "Too deeply nested".
  else \@enumdepth :=L \@enumdepth + 1
       \@enumctr :=L eval(enum@\romannumeral\the\@enumdepth)
       \list{\label(\@enumctr)}
            {\usecounter{\@enumctr}
             \makelabel{LABEL} ==  \hss \llap{LABEL}}
fi
END

\endenumerate == \endlist
```

End of historical LaTeX 2.09 comments.

`\@enumdepth`

```
\newcount\@enumdepth \@enumdepth = 0
```

(*End definition for* `\@enumdepth`*. This function is documented on page* **??**.)

`\c@enumi`
`\c@enumii`
`\c@enumii`
`\c@enumiv`

```
\@definecounter{enumi}
\@definecounter{enumii}
\@definecounter{enumiii}
\@definecounter{enumiv}
```

(*End definition for* `\c@enumi` *and others. These functions are documented on page* **??**.)

`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
```

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

```
\itemize ==
  BEGIN
    if \@itemdepth > 3
      then  errormessage: 'Too deeply nested'.
      else \@itemdepth :=L \@itemdepth + 1
           \@itemitem  == eval(labelitem\romannumeral\the\@itemdepth)
           \list{\@nameuse{\@itemitem}}
                 {\makelabel{LABEL} ==  \hss \llap{LABEL}}
    fi
  END

\enditemize ==  \endlist
```

End of historical LaTeX 2.09 comments.

`\@itemdepth`

```
\newcount\@itemdepth \@itemdepth = 0
```

(*End definition for* `\@itemdepth`*. This function is documented on page* **??**.)

`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 G
ltboxes.dtx

1 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 (.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\makebox}{Make \makebox robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\makebox{%
  \leavevmode
  \@ifnextchar(%)
    \@makepicbox
    {\@ifnextchar[\@makebox\mbox}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\makebox}{Make \makebox robust}%
```

```
⟨latexrelease⟩\def\makebox{%
⟨latexrelease⟩   \leavevmode
⟨latexrelease⟩   \@ifnextchar(%)
⟨latexrelease⟩     \@makepicbox
⟨latexrelease⟩     {\@ifnextchar[\@makebox\mbox}}%
⟨latexrelease⟩\expandafter\let\csname makebox \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\makebox`*. This function is documented on page* **??**.)

`\mbox` The basic horizontal box command for LaTeX.

```
\DeclareRobustCommand\mbox[1]{\leavevmode\hbox{#1}}
```

(*End definition for* `\mbox`*. This function is documented on page* **??**.)

`\@makebox` Look for a possible second optional argument (defaults to `c`).

```
\def\@makebox[#1]{%
  \@ifnextchar [{\@imakebox[#1]}{\@imakebox[#1][c]}}
```

(*End definition for* `\@makebox`*. This function is documented on page* **??**.)

`\@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 definition for* `\@begin@tempboxa`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@end@tempboxa`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\bm@c` *and others. These functions are documented on page* **??**.)

`\@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}
```

(*End definition for* `\@imakebox`*. This function is documented on page* **??**.)

`\@makepicbox` Picture mode form of `\makebox`.

```
\def\@makepicbox(#1,#2){%
  \@ifnextchar[{\@imakepicbox(#1,#2)}{\@imakepicbox(#1,#2)[]}}
```

(*End definition for* `\@makepicbox`*. This function is documented on page* **??**.)

`\@imakepicbox` picture mode version

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\@imakepicbox}{default units}%
\long\def\@imakepicbox(#1,#2)[#3]#4{%
  \@defaultunitsset\@tempdimc{#2}\unitlength
  \vbox to\@tempdimc
   {\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
    \@defaultunitsset\@tempdimc{#1}\unitlength
    \hb@xt@\@tempdimc{\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 LaTeX Book. The kern is removed in compatibility mode.

```
    \kern\z@}}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@imakepicbox}{default units}%
⟨latexrelease⟩\long\def\@imakepicbox(#1,#2)[#3]#4{%
⟨latexrelease⟩  \vbox to#2\unitlength
⟨latexrelease⟩   {\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
⟨latexrelease⟩    \let\mb@t\vss
⟨latexrelease⟩    \@tfor\reserved@a :=#3\do{%
⟨latexrelease⟩      \if s\reserved@a
⟨latexrelease⟩        \let\mb@l\relax\let\mb@r\relax
⟨latexrelease⟩      \else
⟨latexrelease⟩        \expandafter\let\csname mb@\reserved@a\endcsname\relax
⟨latexrelease⟩      \fi}%
⟨latexrelease⟩    \mb@t
⟨latexrelease⟩    \hb@xt@ #1\unitlength{\mb@l #4\mb@r}%
⟨latexrelease⟩    \mb@b
⟨latexrelease⟩    \kern\z@}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@imakepicbox`. *This function is documented on page* **??**.)

`\set@color` This macro is initially a no-op, but the colour package will redefine it to insert a `\special`.

```
\let\set@color\relax
```

(*End definition for* `\set@color`. *This function is documented on page* **??**.)

`\color@begingroup` `\color@endgroup` `\color@setgroup` `\normalcolor` `\color@hbox` `\color@vbox` `\color@endbox` These macros are initially a no-op, but the colour package will redefine them to be `\begingroup`, `\endgroup`, `\begingroup\set@color`, `\hbox\bgroup\color@begingroup`, `\color@endgroup\egroup`. and ⟨*set to main document colour*⟩ respectively.

```
\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
```

(*End definition for* `\color@begingroup` *and others. These functions are documented on page* **??**.)

`\newsavebox` Allocate a new 'savebox'.

```
\def\newsavebox#1{\@ifdefinable{#1}{\newbox#1}}
```

(*End definition for* `\newsavebox`. *This function is documented on page* **??**.)

`\savebox` Save `#1` in a box register.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\savebox}{Make \savebox robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\savebox[1]{%
  \@ifnextchar(%)
    {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\savebox}{Make \savebox robust}%
⟨latexrelease⟩\def\savebox#1{%
⟨latexrelease⟩  \@ifnextchar(%)
⟨latexrelease⟩    {\@savepicbox#1}{\@ifnextchar[{\@savebox#1}{\sbox#1}}}%
⟨latexrelease⟩\expandafter\let\csname savebox \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\savebox`. *This function is documented on page* **??**.)

`\sbox` Save `#1` in a box register.

```
\DeclareRobustCommand\sbox[2]{\setbox#1\hbox{%
  \color@setgroup#2\color@endgroup}}
```

(*End definition for* `\sbox`. *This function is documented on page* **??**.)

\@savebox Look for second optional argument.

```
\def\@savebox#1[#2]{%
  \@ifnextchar [{\@isavebox#1[#2]}{\@isavebox#1[#2][c]}}
```

(*End definition for* \@savebox*. This function is documented on page* ??.)

\@isavebox

```
\long\def\@isavebox#1[#2][#3]#4{%
  \sbox#1{\@imakebox[#2][#3]{#4}}}
```

(*End definition for* \@isavebox*. This function is documented on page* ??.)

\@savepicbox Picture mode version of \savebox.

```
\def\@savepicbox#1(#2,#3){%
  \@ifnextchar[%]
    {\@isavepicbox#1(#2,#3)}{\@isavepicbox#1(#2,#3)[]}}
```

(*End definition for* \@savepicbox*. This function is documented on page* ??.)

\@isavepicbox Picture mode version of \savebox.

```
\long\def\@isavepicbox#1(#2,#3)[#4]#5{%
  \sbox#1{\@imakepicbox(#2,#3)[#4]{#5}}}
```

(*End definition for* \@isavepicbox*. This function is documented on page* ??.)

\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}
```

(*End definition for* \lrbox*. This function is documented on page* ??.)

\endlrbox End the lrbox environment.

```
\def\endlrbox{\unskip\color@endgroup}
```

(*End definition for* \endlrbox*. This function is documented on page* ??.)

\usebox unchanged

```
\DeclareRobustCommand\usebox[1]{\leavevmode\copy #1\relax}
```

(*End definition for* \usebox*. This function is documented on page* ??.)

\frame The following definition of \frame was written by Pavel Curtis (Extra space removed 14 Jan 88) RmS 92/08/24: Replaced occurrence of \@halfwidth by \@wholewidth

```
\DeclareRobustCommand\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}}
```

(*End definition for* \frame*. This function is documented on page* ??.)

\fboxrule \fboxsep user level parameters,

```
\newdimen\fboxrule
\newdimen\fboxsep
```

(*End definition for* \fboxrule *and* \fboxsep*. These functions are documented on page* ??.)

\fbox Abbreviated framed box command.

```
\DeclareRobustCommand\fbox[1]{%
  \leavevmode
  \setbox\@tempboxa\hbox{%
    \color@begingroup
      \kern\fboxsep{#1}\kern\fboxsep
    \color@endgroup}%
  \@frameb@x\relax}
```

(*End definition for* \fbox*. This function is documented on page* ??.)

\framebox Framed version of \makebox.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\framebox}{Make \framebox robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\framebox{%
  \@ifnextchar(%)
    \@framepicbox{\@ifnextchar[\@framebox\fbox}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\framebox}{Make \framebox robust}%
⟨latexrelease⟩\def\framebox{%
⟨latexrelease⟩  \@ifnextchar(%)
⟨latexrelease⟩    \@framepicbox{\@ifnextchar[\@framebox\fbox}}%
⟨latexrelease⟩\expandafter\let\csname framebox \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\framebox`*. This function is documented on page* **??**.)

`\@framebox` Deal with optional arguments.

```
\def\@framebox[#1]{%
  \@ifnextchar[%]
    {\@iframebox[#1]}%
    {\@iframebox[#1][c]}}
```

(*End definition for* `\@framebox`*. This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@iframebox`*. This function is documented on page* **??**.)

`\@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}%
                          }%
        }%
}
```

(*End definition for* `\@frameb@x`*. This function is documented on page* **??**.)

`\@framepicbox` Picture mode version.

```
\def\@framepicbox(#1,#2){%
  \@ifnextchar[{\@iframepicbox(#1,#2)}{\@iframepicbox(#1,#2)[]}}
```

(*End definition for* `\@framepicbox`*. This function is documented on page* **??**.)

\@iframepicbox Picture mode version.

```
\long\def\@iframepicbox(#1,#2)[#3]#4{%
  \frame{\@imakepicbox(#1,#2)[#3]{#4}}}
```

(*End definition for* \@iframepicbox*. This function is documented on page* **??**.)

\parbox The main vertical-box command for LaTeX.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\parbox}{Make \parbox robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\parbox{%
  \@ifnextchar[%]
    \@iparbox
    {\@iiiparbox c\relax[s]}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\parbox}{Make \parbox robust}%
⟨latexrelease⟩\def\parbox{%
⟨latexrelease⟩  \@ifnextchar[%]
⟨latexrelease⟩    \@iparbox
⟨latexrelease⟩    {\@iiiparbox c\relax[s]}}%
⟨latexrelease⟩\expandafter\let\csname parbox \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \parbox*. This function is documented on page* **??**.)

\@iparbox Optional argument handling.

```
\def\@iparbox[#1]{%
  \@ifnextchar[%]
    {\@iiparbox{#1}}%
    {\@iiiparbox{#1}\relax[s]}}
```

(*End definition for* \@iparbox*. This function is documented on page* **??**.)

\@iiparbox Optional argument handling.

```
\def\@iiparbox#1[#2]{%
  \@ifnextchar[%]
    {\@iiiparbox{#1}{#2}}%
    {\@iiiparbox{#1}{#2}[#1]}}
```

(*End definition for* \@iiparbox*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* `\@iiiparbox` *and* `\@parboxto`*. These functions are documented on page* **??**.)

`\@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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017-04-15}%
⟨latexrelease⟩                 {\normallineskiplimit}
⟨latexrelease⟩                 {reset \lineskiplimit}%
⟨*2ekernel | latexrelease⟩
\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
  \lineskiplimit\normallineskiplimit
  \baselineskip\normalbaselineskip
  \sloppy}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000-00-00}%
⟨latexrelease⟩                 {\normallineskiplimit}
⟨latexrelease⟩                 {reset \lineskiplimit}%
⟨latexrelease⟩\def\@arrayparboxrestore{%
⟨latexrelease⟩  \let\if@nobreak\iffalse
⟨latexrelease⟩  \let\if@noskipsec\iffalse
```

```
⟨latexrelease⟩  \let\par\@@par
⟨latexrelease⟩  \let\-\@dischyph
⟨latexrelease⟩  \let\'\@acci\let\`\@accii\let\=\@acciii
⟨latexrelease⟩  \parindent\z@ \parskip\z@skip
⟨latexrelease⟩  \everypar{}%
⟨latexrelease⟩  \linewidth\hsize
⟨latexrelease⟩  \@totalleftmargin\z@
⟨latexrelease⟩  \leftskip\z@skip \rightskip\z@skip \@rightskip\z@skip
⟨latexrelease⟩  \parfillskip\@flushglue \lineskip\normallineskip
⟨latexrelease⟩  \baselineskip\normalbaselineskip
⟨latexrelease⟩  \sloppy}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@arrayparboxrestore`*. This function is documented on page* **??**.)

`\parboxrestore` Restore various paragraph parameters, and also `\\`.

```
\def\@parboxrestore{\@arrayparboxrestore\let\\\@normalcr}
```

(*End definition for* `\parboxrestore`*. This function is documented on page* **??**.)

`\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
```

(*End definition for* `\if@minipage`*. This function is documented on page* **??**.)

`\minipage` Essentially an environment form of `\parbox`.

```
\def\minipage{%
  \@ifnextchar[%]
    \@iminipage
    {\@iiiminipage c\relax[s]}}
```

(*End definition for* `\minipage`*. This function is documented on page* **??**.)

`\@iminipage` Optional argument handling.

```
\def\@iminipage[#1]{%
  \@ifnextchar[%]
    {\@iiminipage{#1}}%
    {\@iiiminipage{#1}\relax[s]}}
```

(*End definition for* `\@iminipage`*. This function is documented on page* **??**.)

`\@iiminipage` Optional argument handling.

```
\def\@iiminipage#1[#2]{%
  \@ifnextchar[%]
    {\@iiiminipage{#1}{#2}}%
    {\@iiiminipage{#1}{#2}[#1]}}
```

(*End definition for* `\@iiminipage`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* \@iiiminipage. *This function is documented on page* ??.)

\@minipagerestore Hook so that other styles can reset other commands in a minipage.

```
 \let\@minipagerestore=\relax
```

(*End definition for* \@minipagerestore. *This function is documented on page* ??.)

\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}}
```

(*End definition for* \endminipage. *This function is documented on page* ??.)

\@mplistdepth
\@mpfootins Versions of \@listdepth and \footins local to minipage.

```
\newcount\@mplistdepth
\newinsert\@mpfootins
```

(*End definition for* \@mplistdepth *and* \@mpfootins. *These functions are documented on page* ??.)

\@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}}
```

(*End definition for* `\@mpfootnotetext`*. This function is documented on page* **??**.)

```
\newif\if@pboxsw
```

`\rule` Draw a rule of the specified size.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\rule}{Make \rule robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\rule}{Make \rule robust}%
⟨latexrelease⟩\def\rule{\@ifnextchar[\@rule{\@rule[\z@]}}%
⟨latexrelease⟩\expandafter\let\csname rule \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\rule`*. This function is documented on page* **??**.)

`\@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}}
```

(*End definition for* `\@rule`*. This function is documented on page* **??**.)

`\@@underline` Saved primitive `\underline`.

```
\let\@@underline\underline
```

(*End definition for* `\@@underline`*. This function is documented on page* **??**.)

`\underline` LaTeX version works outside math.

```
\DeclareRobustCommand\underline[1]{%
  \relax
  \ifmmode\@@underline{#1}%
  \else $\@@underline{\hbox{#1}}\m@th$\relax\fi}
```

(*End definition for* `\underline`*. This function is documented on page* **??**.)

\raisebox Raise a box, and change its vertical dimensions.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\raisebox}{Make \raisebox robust}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand\raisebox[1]{%
  \leavevmode
  \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\raisebox}{Make \raisebox robust}%
⟨latexrelease⟩\def\raisebox#1{%
⟨latexrelease⟩  \leavevmode
⟨latexrelease⟩  \@ifnextchar[{\@rsbox{#1}}{\@irsbox{#1}[]}}
⟨latexrelease⟩\expandafter\let\csname raisebox \endcsname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\raisebox`*. This function is documented on page* **??**.)

\@rsbox Optional argument handling.

```
\def\@rsbox#1[#2]{%
  \@ifnextchar[{\@iirsbox{#1}[#2]}{\@irsbox{#1}[#2]}}
```

(*End definition for* `\@rsbox`*. This function is documented on page* **??**.)

\@argrsbox ...

(*End definition for* `\@argrsbox`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* `\@irsbox`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* `\@iirsbox`*. This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@finalstrut`*. This function is documented on page* **??**.)

1.1 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}
\DeclareRobustCommand\leftline[1]{\@@line{#1\hss}}
\DeclareRobustCommand\rightline[1]{\@@line{\hss#1}}
\DeclareRobustCommand\centerline[1]{\@@line{\hss#1\hss}}
```

(*End definition for* `\leftline` *and others. These functions are documented on page* **??**.)

`\rlap` `\llap` `\clap` These macros place text to the left or right of the current reference point without taking up space.

```
\DeclareRobustCommand\rlap[1]{\hb@xt@\z@{#1\hss}}
\DeclareRobustCommand\llap[1]{\hb@xt@\z@{\hss#1}}
```

And here is the version that centers, it was initially introduced by `mathtools`.

```
\DeclareRobustCommand\clap[1]{\hb@xt@\z@{\hss#1\hss}}
```

(*End definition for* `\rlap`*,* `\llap`*, and* `\clap`*. These functions are documented on page* **??**.)

⟨/2ekernel⟩

File H
lttab.dtx

1 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.

1.1 tabbing

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

```
\dimen(\@firsttab + i) = distance of tab stop i from left margin
       0 <= i <= 15 (?).

\dimen\@firsttab is initialized to \@totalleftmargin, so it starts
    at the prevailing left margin.

\@maxtab          = number of highest defined tab register
                      probably = \@firsttab + 12
\@nxttabmar = tab stop number of next line's left margin
\@curtabmar = tab stop number of current line's left margin
\@curtab    = number of the current tab. At start of line,
                    it equals \@curtabmar
\@hightab   = largest tab number currently defined.
\@tabpush   = depth of \pushtab's

\box\@curline     = contents of current line, excluding left margin
                      skip, and excluding contents of current field
\box\@curfield    = contents of current field

@rjfield          = switch: T iff the last field of the line should
                      be right-justified at the right margin.

\tabbingsep           = distance left by the \' command between the
                      current position and the field that is
                      "left-shifted".

UTILITY MACROS
 \@stopfield  : closes the current field
 \@addfield   : adds the current field to the current line.
 \@contfield  : continues the current field
 \@startfield : begins the next field
 \@stopline   : closes the current line and outputs it
```

```
  \@startline  : starts the next line
  \@ifatmargin : an \if that is true iff the current line.
                 has width zero

\@startline ==
 BEGIN
  \@curtabmar :=G \@nxttabmar
  \@curtab :=G \@curtabmar
  \box\@curline :=G null
  \@startfield
  \strut
 END

\@stopline ==
 BEGIN
  \unskip
  \@stopfield
  if @rjfield = T
    then  @rjfield :=G F
          \@tempdima := \@totalleftmargin + \linewidth
          \hb@xt@ \@tempdima{\@itemfudge
                             \hskip \dimen\@curtabmar
                             \box\@curline
                             \hfil
                             \box\@curfield}
    else \@addfield
         \hbox {\@itemfudge
                \hskip \dimen\@curtabmar
                \box\@curline}
  fi
 END

\@startfield ==
 BEGIN
   \box\@curfield :=G \hbox {
 END

\@stopfield ==
 BEGIN
    }
 END

\@contfield ==
 BEGIN
  \box\@curfield :=G \hbox { \unhbox\@currfield  %%} brace matching
 END
\@addfield ==
 BEGIN
  \box\@curline :=G \unbox\@curline * \unbox\@curfield
 END
```

```
\@ifatmargin ==
 BEGIN
  if  dim of box\@curline = 0pt  then
 END


\tabbing ==
 BEGIN
  \lineskip :=L 0pt
  \> == \@rtab
  \< == \@ltab
  \= == \@settab
  \+ == \@tabplus
  \- == \@tabminus
  \‘ == \@tabrj
  \’ == \@tablab
  \\ == BEGIN \@stopline \@startline END
  \\[DIST] == BEGIN
                \@stopline \vskip DIST \@startline\ignorespaces END
  \\* == BEGIN \@stopline \penalty 10000 \@startline END
  \\*[DIST] == BEGIN \@stopline \penalty 10000 \vskip DIST
                     \@startline\ignorespaces              END
  \@hightab :=  \@nxttabmar :=G \@firsttab
  \@tabpush :=G 0
  \dimen\@firsttab := \@totalleftmargin
  @rjfield :=G F
  \trivlist  \item\relax
  if @minipage = F then \vskip \parskip fi
  \box\@tabfbox = \rlap{\indent\the\everypar}
                        % note: \the\everypar sets @inlabel :=G F
  \@itemfudge == BEGIN \box\@tabfbox END
  \@startline
  \ignorespaces
 END

\@endtabbing ==
 BEGIN
  \@stopline
  if \@tabpush > 0 then error message: ”unmatched \poptabs’’ fi
  \endtrivlist
 END

\@rtab ==
 BEGIN
  \@stopfield
  \@addfield
  if \@curtab < \@hightab
    then \@curtab :=G \@curtab + 1
    else error message “Undefined Tab”   fi
```

```
  \@tempdima := \dimen\@curtab - \dimen\@curtabmar
                        - width of box \@curline
  \box\@curline :=G \hbox{\unhbox\@curline + \hskip\@tempdima}
  \@startfield
 END

\@settab ==
 BEGIN
  \@stopfield
  \@addfield
  if \@curtab < \@maxtab
    then \@curtab :=G \@curtab+1
    else error message: "Too many tabs"     fi
  if \@curtab > \@hightab
    then \@hightab :=L \@curtab     fi
  \dimen\@curtab :=L \dimen\@curtabmar + width of \box\@curline
  \@startfield
 END

\@ltab ==
 BEGIN
  \@ifatmargin
    then if \@curtabmar > \@firsttab
            then \@curtab :=G \@curtab - 1
                 \@curtabmar :=G \@curtabmar - 1
            else error message "Too many 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
```

End of historical LaTeX 2.09 comments.

\a The accents \‘ , \’ , and \= that have been redefined inside a tabbing environment can be called by typing \a‘ , \a’ , and \a=. The macro \a is defined in ltoutenc.dtx.

(*End definition for* \a. *This function is documented on page* ??.)

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⟩
\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
```

(*End definition for* `\@firsttab` *and* `\@maxtab`*. These functions are documented on page* **??**.)

`\@nxttabmar`
`\@curtabmar`
`\@curtab`
`\@hightab`
`\@tabpush`

```
\newcount\@nxttabmar
\newcount\@curtabmar
\newcount\@curtab
\newcount\@hightab
\newcount\@tabpush
```

(*End definition for* `\@nxttabmar` *and others. These functions are documented on page* **??**.)

`\@curline`
`\@curfield`
`\@tabfbox`

```
\newbox\@curline
\newbox\@curfield
\newbox\@tabfbox
```

(*End definition for* `\@curline`*,* `\@curfield`*, and* `\@tabfbox`*. These functions are documented on page* **??**.)

`\if@rjfield`

```
\newif\if@rjfield
```

(*End definition for* `\if@rjfield`*. This function is documented on page* **??**.)

`\@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.

```
\def\@startline{%
     \ifnum \@nxttabmar >\@hightab
       \@badtab
       \global\@nxttabmar \@hightab
     \fi
     \global\@curtabmar \@nxttabmar
     \global\@curtab \@curtabmar
     \global\setbox\@curline \hbox {}%
     \@startfield
     \strut}
```

(*End definition for* `\@startline`*. This function is documented on page* **??**.)

`\@stopline`

```
\def\@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}
```

(*End definition for* `\@stopline`*. This function is documented on page* **??**.)

`\@startfield`

```
\def\@startfield{%
  \global\setbox\@curfield\hbox\bgroup\color@begingroup}
```

(*End definition for* `\@startfield`*. This function is documented on page* **??**.)

`\@stopfield`

```
\def\@stopfield{%
  \color@endgroup\egroup}
```

(*End definition for* `\@stopfield`*. This function is documented on page* **??**.)

`\@contfield`

```
\def\@contfield{%
  \global\setbox\@curfield\hbox\bgroup\color@begingroup
  \unhbox\@curfield}
```

(*End definition for* `\@contfield`*. This function is documented on page* **??**.)

`\@addfield`

```
\def\@addfield{\global\setbox\@curline\hbox{\unhbox
      \@curline\unhbox\@curfield}}
```

(*End definition for* `\@addfield`*. This function is documented on page* **??**.)

`\@ifatmargin`

```
\def\@ifatmargin{\ifdim \wd\@curline =\z@}
```

(*End definition for* `\@ifatmargin`*. This function is documented on page* **??**.)

`\@tabcr`

```
\def\@tabcr{\@stopline \@ifstar{\penalty \@M \@xtabcr}\@xtabcr}
```

(*End definition for* `\@tabcr`*. This function is documented on page* **??**.)

`\@xtabcr`

```
\def\@xtabcr{\@ifnextchar[\@itabcr{\@startline\ignorespaces}}
```

(*End definition for* `\@xtabcr`*. This function is documented on page* **??**.)

`\@itabcr`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@itabcr}{Tabbing calc syntax}%
\def\@itabcr[#1]{\@vspace@calcify{#1}\@startline\ignorespaces}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@itabcr}{Tabbing calc syntax}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@itabcr[#1]{\vskip #1\@startline\ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

tabbing We use \relax to prevent \item from scanning too far.

\tabbing
```
\def\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
```
\def\endtabbing{%
  \@stopline\ifnum\@tabpush >\z@ \@badpoptabs \fi\endtrivlist}
```

Omitted \global added to \@rtab 17 Jun 86

\@rtab
```
\def\@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
```
\def\@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
```
\def\@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
```
\def\@tabplus{%
  \ifnum\@nxttabmar<\@hightab
```

```
    \global\advance\@nxttabmar\@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
```

\@tabminus

```
\def\@tabminus{%
  \ifnum\@nxttabmar>\@firsttab
    \global\advance\@nxttabmar\m@ne
  \else
    \@badtab
  \fi
  \ignorespaces}
```

\@tabrj

```
\def\@tabrj{%
  \@stopfield\@addfield\global\@rjfieldtrue\@startfield\ignorespaces}
```

\setbox\@curline made \global in \@tablab. 17 Jun 86

\@tablab

```
\def\@tablab{%
  \@stopfield
  \global\setbox\@curline\hbox{%
    \box\@curline
    \hskip-\wd\@curfield \hskip-\tabbingsep
    \box\@curfield
    \hskip\tabbingsep}%
  \@startfield
  \ignorespaces}
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\pushtabs}{Make commands robust}%
```

\pushtabs

```
\DeclareRobustCommand\pushtabs{%
  \@stopfield\@addfield\global\advance\@tabpush \@ne \begingroup
       \@contfield}
```

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.

\poptabs

```
\DeclareRobustCommand\poptabs{\@stopfield\@addfield
  \ifnum \@tabpush >\z@
    \endgroup
    \global\advance\@tabpush \m@ne
    \ifnum \@curtab >\@hightab
      \global \@curtab \@hightab
      \@badtab
    \fi
  \else
    \@badpoptabs
  \fi
  \@contfield}
```

```
\DeclareRobustCommand\kill{\@stopfield\@startline\ignorespaces}
```

(*End definition for* `\@itabcr` *and others. These functions are documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\pushtabs}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\pushtabs
⟨latexrelease⟩\kernel@make@fragile\poptabs
⟨latexrelease⟩\kernel@make@fragile\kill
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\tabbingsep`

```
\newdimen\tabbingsep
```

(*End definition for* `\tabbingsep`*. This function is documented on page* ??.)

1.2 array and tabular environments

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

```
ARRAY PARAMETERS:
 \arraycolsep
      : half the width separating columns in an array environment
 \tabcolsep
      : half the width separating columns in a tabular environment
 \arrayrulewidth
      : width of rules
 \doublerulesep
      : space between adjacent rules in array or tabular
 \arraystretch
      : line spacing in array and tabular environments is done by
        placing a strut in every row of height and depth
        \arraystretch times the height and depth of the strut
        produced by an ordinary \strut command.

PREAMBLE:
 The PREAMBLE argument of an array or tabular environment can
 contain the following:
   l,r,c   : indicate where entry is to be placed.
   |       : for vertical rule
   @{EXP} : inserts the text EXP in every column.
            \arraycolsep or \tabcolsep  spacing is suppressed.
   *{N}{PRE} : equivalent to writing N copies of PRE in the preamble.
               PRE may contain *{N'}{EXP'} expressions.
   p{LEN} : makes entry in parbox of width LEN.

SPECIAL ARRAY COMMANDS:
```

```
\multicolumn{N}{FORMAT}{ITEM} : replaces the next N column
 items by ITEM, formatted according to FORMAT.
 FORMAT should contain at most one l,r or c.
 If it contains none, then ITEM is ignored.

\vline : draws a vertical line the height of the current row.  May
         appear in an array element entry.
\hline : draws a horizontal line between rows.  Must appear either
         before the first entry (to appear above the first row) or
          right after a \\ command.  If followed by another \hline,
          then adds a \vskip of \doublerulesep.

\cline{i-j} : draws horizontal lines between rows covering columns
              i through j, inclusive.  Multiple commands may follow
              one another to provide lines covering several disjoint
              columns
\extracolsep{WIDTH} : for use inside an @ in the preamble.  Causes
            a WIDTH space to be added between columns for the rest
             of the columns.  This is in addition to the ordinary
             intercolumn space.

\array ==
  BEGIN
    \@acol    == \@arrayacol
    \@classz  == \@arrayclassz
    \@classiv == \@arrayclassiv
    \\        == \@arraycr
    \@halignto == NULL
    \@tabarray
  END

\endarray{NAME} ==  BEGIN  \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.

End of historical LaTeX 2.09 comments.

\extracolsep This command needs to expand during the tabular preamble construction so can't be robust.

```
\def\extracolsep#1{\tabskip #1\relax}
```

(*End definition for* \extracolsep*. This function is documented on page* ??.)

\array

```
\def\array{\let\@acol\@arrayacol \let\@classz\@arrayclassz
 \let\@classiv\@arrayclassiv
 \let\\\@arraycr\let\@halignto\@empty\@tabarray}
```

(*End definition for* \array*. This function is documented on page* ??.)

\endarray
\endtabular
\endtabular*

```
\def\endarray{\crcr\egroup\egroup}
\def\endtabular{\crcr\egroup\egroup $\egroup}
\expandafter \let \csname endtabular*\endcsname = \endtabular
```

(*End definition for* \endarray*,* \endtabular*, and* \endtabular**. These functions are documented on page* ??.)

\tabular

```
\def\tabular{\let\@halignto\@empty\@tabular}
```

(*End definition for* \tabular*. This function is documented on page* ??.)

\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}
```

(*End definition for* \tabular**. This function is documented on page* ??.)

\@tabular

```
\def\@tabular{\leavevmode \hbox \bgroup $\let\@acol\@tabacol
   \let\@classz\@tabclassz
   \let\@classiv\@tabclassiv \let\\\@tabularcr\@tabarray}
```

(*End definition for* \@tabular*. This function is documented on page* ??.)

\@tabarray RmS 91/11/04 added \m@th.

```
\def\@tabarray{\m@th\@ifnextchar[\@array{\@array[c]}}
```

(*End definition for* \@tabarray*. This function is documented on page* ??.)

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}
```

(*End definition for* \@array*. This function is documented on page* ??.)

\@arraycr Array version of \\.

```
\def\@arraycr{%
  ${\ifnum0=`}\fi\@ifstar\@xarraycr\@xarraycr}
```

(*End definition for* \@arraycr*. This function is documented on page* ??.)

\@arraycr

```
\def\@xarraycr{\@ifnextchar[\@argarraycr{\ifnum0=`{\fi}${}\cr}}
```

(*End definition for* \@arraycr*. This function is documented on page* ??.)

\@argarraycr

```
\def\@argarraycr[#1]{%
  \ifnum0=`{\fi}${}\ifdim #1>\z@ \@xargarraycr{#1}\else
   \@yargarraycr{#1}\fi}
```

(*End definition for* `\@argarraycr`*. This function is documented on page* **??**.)

\tabularnewline Tabular version of \\.

```
\let\tabularnewline\relax
```

(*End definition for* `\tabularnewline`*. This function is documented on page* **??**.)

\@tabularcr

```
\def\@tabularcr{%
  {\ifnum0=`}\fi\@ifstar\@xtabularcr\@xtabularcr}
```

(*End definition for* `\@tabularcr`*. This function is documented on page* **??**.)

\@xtabularcr

```
\def\@xtabularcr{\@ifnextchar[\@argtabularcr{\ifnum0=`{\fi}\cr}}
```

(*End definition for* `\@xtabularcr`*. This function is documented on page* **??**.)

\@argtabularcr

```
\def\@argtabularcr[#1]{%
  \ifnum0=`{\fi}%
    \ifdim #1>\z@
      \unskip\@xargarraycr{#1}%
    \else
      \@yargarraycr{#1}%
    \fi}
```

(*End definition for* `\@argtabularcr`*. This function is documented on page* **??**.)

\@xargarraycr

```
\def\@xargarraycr#1{\@tempdima #1\advance\@tempdima \dp \@arstrutbox
   \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr}
```

(*End definition for* `\@xargarraycr`*. This function is documented on page* **??**.)

\@yargarraycr

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@yargarraycr}{tabular support calc syntax}%
\def\@yargarraycr#1{\cr\noalign{\@vspace@calcify{#1}}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@yargarraycr}{tabular support calc syntax}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@yargarraycr#1{\cr\noalign{\vskip #1}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@yargarraycr`*. This function is documented on page* **??**.)

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

```
\multicolumn{NUMBER}{FORMAT}{ITEM} ==
  BEGIN
  \multispan{NUMBER}
  \begingroup
  \@addamp == null
  \@mkpream{FORMAT}
  \@sharp == ITEM
  \protect == \relax
  \@startpbox == \@@startpbox
  \@endpbox == \@@endpbox
  \@arstrut
  \@preamble
  \endgroup
  END
```

End of historical LaTeX 2.09 comments.

The command `\def\@addamp{}` was removed from `\multicolumn` on 6 Dec 86 because it caused embedded array environments not to work. I think that it was included originally to prevent an error message if the 2nd argument to the `\multicolumn` command had two column specifiers.

8 Feb 89 — `\hbox{}` added after `\@preamble` to correct bug that occurred if `\multicolumn` preceded `\\[D]` with `D > 0`, caused by `\\[]` command doing an `\unskip`, which removed `\tabcolsep` glue inserted by `\multicolumn`.

This has been made long so that, for example, a `p`-column can contain multiple paragraphs; maybe the arguments of `@`-expressions should also be able to contain multiple paragraphs.

```
\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}
```

(*End definition for* `\multicolumn`*. This function is documented on page* **??**.)

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

Codes for classes and character numbers of array, tabular and multicolumn arguments.

Character	Class	Number
c	0	0
l	0	1
r	0	2
|	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
```

End of historical LaTeX 2.09 comments.

\@xexnoop

```
\def\@xexnoop #1\@@{}
```

(*End definition for* \@xexnoop*. This function is documented on page* ??.)

\@expast

```
\def\@expast#1{\@xexpast #1*0x\@@}
```

(*End definition for* \@expast*. This function is documented on page* ??.)

\@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\@@}
```

(*End definition for* `\@xexpast`*. This function is documented on page* **??**.)

\if@firstamp
\@addamp

```
\newif\if@firstamp
```

```
\def\@addamp{%
  \if@firstamp
    \@firstampfalse
  \else
    \edef\@preamble{\@preamble &}%
  \fi}
```

(*End definition for* `\if@firstamp` *and* `\@addamp`*. These functions are documented on page* **??**.)

\@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}
```

(*End definition for* `\@arrayacol` *and others. These functions are documented on page* **??**.)

\@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}
```

(*End definition for* `\@mkpream`*. This function is documented on page* **??**.)

\@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}}
```

(*End definition for* `\@arrayclassz`*. This function is documented on page* **??**.)

\@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

        \hskip1sp%
        \ignorespaces\@sharp\unskip\hfil
      \or

        \hskip1sp\ignorespaces\@sharp\unskip\hfil
      \or
        \hfil\hskip1sp\ignorespaces\@sharp\unskip
      \fi}}}
```

(*End definition for* `\@tabclassz`*. This function is documented on page* **??**.)

\@classi

```
\def\@classi{%
  \ifcase\@lastchclass
    \@acol\@arrayrule
  \or
    \@addtopreamble{\hskip \doublerulesep}\@arrayrule
  \or
  \or
  \or
    \@arrayrule
  \or
    \@acol\@arrayrule
  \or
```

```
    \@arrayrule
  \fi}
```

(*End definition for* `\@classi`. *This function is documented on page* **??**.)

`\@classii`

```
\def\@classii{%
  \ifcase\@lastchclass
  \or
    \@addtopreamble{\hskip .5\arrayrulewidth}%
  \fi}
```

(*End definition for* `\@classii`. *This function is documented on page* **??**.)

`\@classiii`

```
\def\@classiii{\ifcase \@lastchclass \@acolampacol \or
   \@addamp\@acol \or
   \or \or \@addamp \or
   \@acolampacol \or \@ampacol \fi}
```

(*End definition for* `\@classiii`. *This function is documented on page* **??**.)

`\@tabclassiv`

```
\def\@tabclassiv{\@addtopreamble\@nextchar}
```

(*End definition for* `\@tabclassiv`. *This function is documented on page* **??**.)

`\@arrayclassiv`

```
\def\@arrayclassiv{\@addtopreamble{$\@nextchar$}}
```

(*End definition for* `\@arrayclassiv`. *This function is documented on page* **??**.)

`\@classv`

```
\def\@classv{\@addtopreamble{\@startpbox{\@nextchar}\ignorespaces
\@sharp\@endpbox}}
```

(*End definition for* `\@classv`. *This function is documented on page* **??**.)

`\@addtopreamble`

```
\def\@addtopreamble#1{\edef\@preamble{\@preamble #1}}
```

(*End definition for* `\@addtopreamble`. *This function is documented on page* **??**.)

`\@chclass`
`\@lastchclass`
`\@chnum`

```
\newcount\@chclass
\newcount\@lastchclass
\newcount\@chnum
```

(*End definition for* `\@chclass`, `\@lastchclass`, *and* `\@chnum`. *These functions are documented on page* **??**.)

`\arraycolsep`
`\tabcolsep`
`\arrayrulewidth`
`\doublerulesep`

```
\newdimen\arraycolsep
\newdimen\tabcolsep
\newdimen\arrayrulewidth
\newdimen\doublerulesep
```

(*End definition for* `\arraycolsep` *and others. These functions are documented on page* **??**.)

`\arraystretch`

```
\def\arraystretch{1}    % Default value.
```

(*End definition for* `\arraystretch`*. This function is documented on page* **??**.)

`\@arstrutbox`
`\@arstrut`

```
\newbox\@arstrutbox
```

```
\def\@arstrut{%
  \relax\ifmmode\copy\@arstrutbox\else\unhcopy\@arstrutbox\fi}
```

(*End definition for* `\@arstrutbox` *and* `\@arstrut`*. These functions are documented on page* **??**.)

`\@arrayrule`

```
\def\@arrayrule{\@addtopreamble{\hskip -.5\arrayrulewidth
   \vrule \@width \arrayrulewidth\hskip -.5\arrayrulewidth}}
```

(*End definition for* `\@arrayrule`*. This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@testpatch`*. This function is documented on page* **??**.)

`\hline`

```
\def\hline{%
  \noalign{\ifnum0=`}\fi\hrule \@height \arrayrulewidth \futurelet
   \reserved@a\@xhline}
```

(*End definition for* `\hline`*. This function is documented on page* **??**.)

`\@xhline`

```
\def\@xhline{\ifx\reserved@a\hline
               \vskip\doublerulesep
```

Measure from the middle of the rules.

```
               \vskip-\arrayrulewidth
             \fi
      \ifnum0=`{\fi}}
```

(*End definition for* `\@xhline`*. This function is documented on page* **??**.)

`\vline`

```
\def\vline{\vrule \@width \arrayrulewidth}
```

(*End definition for* `\vline`*. This function is documented on page* **??**.)

\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}}
```

(*End definition for* \cline *and* \@cline*. These functions are documented on page* ??.)

\mscount

The \mscount counter is no longer declared, saving a csname and a register. It is declared in compatibility mode.

(*End definition for* \mscount*. This function is documented on page* ??.)

\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}
```

(*End definition for* \multispan*,* \@multispan*, and* \sp@n*. These functions are documented on page* ??.)

\@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.

(*End definition for* `\@startpbox` *and* `\@endpbox`*. These functions are documented on page* **??**.)

`\@@startpbox`
`\@@endpbox`

```
\let\@@startpbox=\@startpbox
\let\@@endpbox=\@endpbox
```

(*End definition for* `\@@startpbox` *and* `\@@endpbox`*. These functions are documented on page* **??**.)

⟨/2ekernel⟩

File I
ltpictur.dtx

1 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.

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

```
\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 }
                  height of \@picbox := \@picht
                  depth   of \@picbox := 0
                  \mbox{\box\@picbox}   %% change 26 Aug 91
  END

\put(X, Y){OBJ} ==
  BEGIN
```

```
    \@killglue
    \raise Y * \unitlength  \hb@xt@ 0pt { \hskip X * \unitlength
                                          OBJ \hss            }
    \ignorespaces
  END

\multiput(X,Y)(DELX,DELY){N}{OBJ} ==
  BEGIN
   \@killglue
   \@multicnt := N
   \@xdim  := X * \unitlength
   \@ydim  := Y * \unitlength
   while \@multicnt > 0
     do \raise \@ydim \hb@xt@ 0pt { \hskip \@xdim
                                          OBJ \hss   }
        \@multicnt := \@multicnt - 1
        \@xdim     := \@xdim + DELX * \unitlength
        \@ydim     := \@ydim + DELY * \unitlength
     od
   \ignorespaces
  END
```

`\shortstack[POS]{TEXT}` : Makes a `\vbox` containing TEXT stacked as a one-column array, positioned l, r or c as indicated by POS.

End of historical LaTeX 2.09 comments.

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

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

`\@wholewidth`
`\@halfwidth`

```
⟨*2ekernel⟩
\newdimen\@wholewidth
\newdimen\@halfwidth
```

(*End definition for* `\@wholewidth` *and* `\@halfwidth`*. These functions are documented on page* ??.)

`\unitlength`

```
\newdimen\unitlength \unitlength =1pt
```

(*End definition for* `\unitlength`*. This function is documented on page* ??.)

`\@picbox`
`\@picht`

```
\newbox\@picbox
\newdimen\@picht
```

(*End definition for* `\@picbox` *and* `\@picht`*. These functions are documented on page* ??.)

`\@defaultunitsset` Set a length register, `#1`, accepting number or an etex length expression, `#2`, with default unit, `#3`.

The register name in `#1` can be prefixed by `\advance` so that the register is incremented by the supplied value.

`\@defaultunitsset{\advance\@vxx}{\textwidth-15pt}\unitlength`

`#3` can be a literal unit such as `cm` or a length register such as `\unitlength`.

This is used in all `picture` commands that take picture coordinates. So `\put(2,2)` as previously but now `\put(\textwidth-5cm,0.4\texteight)` Note that you can only use expressions with lengths, `\put(1+2,0)` is not supported.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@defaultunitsset}{default units}%
\def\@defaultunitsset#1#2#3{%
  \@defaultunits#1\dimexpr#2#3\relax\relax\@nnil}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@defaultunitsset}{default units}%
⟨latexrelease⟩\let\@defaultunitsset\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@defaultunitsset`*. This function is documented on page* **??**.)

`\picture` `#1` should be white space.

`#1` should be a `(` (eating any white space before the bracket),

`\pictur@`

```
\long\def\picture#1{\pictur@#1}
\def\pictur@(#1){%
  \@ifnextchar({\@picture(#1)}{\@picture(#1)(0,0)}}
```

(*End definition for* `\picture` *and* `\pictur@`*. These functions are documented on page* **??**.)

`\@picture`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@picture}{default units}%
\def\@picture(#1,#2)(#3,#4){%
  \@defaultunitsset\@picht{#2}\unitlength
  \@defaultunitsset\@tempdimc{#1}\unitlength
  \setbox\@picbox\hb@xt@\@tempdimc\bgroup
    \@defaultunitsset\@tempdimc{#3}\unitlength
    \hskip -\@tempdimc
    \@defaultunitsset\@tempdimc{#4}\unitlength
    \lower\@tempdimc\hbox\bgroup
      \ignorespaces}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@picture}{default units}%
⟨latexrelease⟩\def\@picture(#1,#2)(#3,#4){%
⟨latexrelease⟩  \@picht#2\unitlength
⟨latexrelease⟩  \setbox\@picbox\hb@xt@#1\unitlength\bgroup
⟨latexrelease⟩    \hskip -#3\unitlength
⟨latexrelease⟩    \lower #4\unitlength\hbox\bgroup
⟨latexrelease⟩      \ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@picture`. *This function is documented on page* ??.)

`\endpicture`

```
\def\endpicture{%
  \egroup\hss\egroup
    \ht\@picbox\@picht\dp\@picbox\z@
    \mbox{\box\@picbox}}
```

(*End definition for* `\endpicture`. *This function is documented on page* ??.)

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).

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\put}{default units}%
⟨latexrelease⟩\expandafter\let\csname put \endcsname\@undefind
\long\def\put(#1,#2)#3{%
  \@killglue
  \@defaultunitsset\@tempdimc{#2}\unitlength
  \raise\@tempdimc
  \hb@xt@\z@{%
    \@defaultunitsset\@tempdimc{#1}\unitlength
    \kern\@tempdimc
    #3\hss}%
  \ignorespaces}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\put}{default units}%
⟨latexrelease⟩\expandafter\let\csname put \endcsname\@undefind
⟨latexrelease⟩\long\def\put(#1,#2)#3{%
⟨latexrelease⟩  \@killglue\raise#2\unitlength
⟨latexrelease⟩  \hb@xt@\z@{\kern#1\unitlength #3\hss}%
⟨latexrelease⟩  \ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\multiput` `#3` had better be a `(`.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
```

```
⟨latexrelease⟩                  {\multiput}{default units}%
⟨latexrelease⟩\expandafter\let\csname multiput \endcsname\@undefind
\def\multiput(#1,#2)#3{%
  \@defaultunitsset\@xdim{#1}\unitlength
  \@defaultunitsset\@ydim{#2}\unitlength
   \@multiput(}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\multiput}{default units}%
⟨latexrelease⟩\expandafter\let\csname multiput \endcsname\@undefind
⟨latexrelease⟩\def\multiput(#1,#2)#3{%
⟨latexrelease⟩  \@xdim #1\unitlength
⟨latexrelease⟩  \@ydim #2\unitlength
⟨latexrelease⟩   \@multiput(}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\multiput`*. This function is documented on page* **??**.)

`\@multiput`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@multiput}{default units}%
\long\def\@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
     \@defaultunitsset{\advance\@xdim}{#1}\unitlength
     \@defaultunitsset{\advance\@ydim}{#2}\unitlength}%
  \ignorespaces}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@multiput}{default units}%
⟨latexrelease⟩\long\def\@multiput(#1,#2)#3#4{%
⟨latexrelease⟩  \@killglue\@multicnt #3\relax
⟨latexrelease⟩  \@whilenum \@multicnt >\z@\do
⟨latexrelease⟩    {\raise\@ydim\hb@xt@\z@{\kern\@xdim #4\hss}%
⟨latexrelease⟩     \advance\@multicnt\m@ne
⟨latexrelease⟩     \advance\@xdim#1\unitlength\advance\@ydim#2\unitlength}%
⟨latexrelease⟩  \ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@multiput`*. This function is documented on page* **??**.)

`\@killglue`

```
\def\@killglue{\unskip\@whiledim \lastskip >\z@\do{\unskip}}
```

(*End definition for* `\@killglue`*. This function is documented on page* **??**.)

\thinlines
\thicklines

```
\DeclareRobustCommand\thinlines{\let\@linefnt\tenln
  \let\@circlefnt\tencirc
  \@wholewidth\fontdimen8\tenln \@halfwidth .5\@wholewidth}
\DeclareRobustCommand\thicklines{\let\@linefnt\tenlnw
  \let\@circlefnt\tencircw
  \@wholewidth\fontdimen8\tenlnw \@halfwidth .5\@wholewidth}
```

(*End definition for* `\thinlines` *and* `\thicklines`*. These functions are documented on page* **??**.)

\linethickness

```
\DeclareRobustCommand*\linethickness[1]
   {\@wholewidth #1\relax \@halfwidth .5\@wholewidth \ignorespaces}
```

(*End definition for* `\linethickness`*. This function is documented on page* **??**.)

\ishortstack

```
\def\shortstack{\@ifnextchar[\@shortstack{\@shortstack[c]}}
```

(*End definition for* `\ishortstack`*. This function is documented on page* **??**.)

\@ishortstack

```
\def\@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}
```

(*End definition for* `\@ishortstack`*. This function is documented on page* **??**.)

\@ishortstack

```
\def\@ishortstack#1{\ialign{\mb@l {##}\unskip\mb@r\cr #1\crcr}\egroup}
```

(*End definition for* `\@ishortstack`*. This function is documented on page* **??**.)

\@stackcr
\@ixstackcr

```
\def\@stackcr{\@ifstar\@ixstackcr\@ixstackcr}
\def\@ixstackcr{\@ifnextchar[\@istackcr{\cr\ignorespaces}}
```

(*End definition for* `\@stackcr` *and* `\@ixstackcr`*. These functions are documented on page* **??**.)

\@istackcr

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\@istackcr}{\shortstack calc support}%
\def\@istackcr[#1]{\cr\noalign{\@vspace@calcify{#1}}\ignorespaces}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@istackcr}{\shortstack calc support}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@istackcr[#1]{\cr\noalign{\vskip #1}\ignorespaces}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@istackcr`. *This function is documented on page* ??.)

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

```
\line(X,Y){LEN} ==
BEGIN
 \@xarg    := X
 \@yarg    := Y
 \@linelen := LEN * \unitlength
 if \@xarg = 0
    then \@vline
    else if \@yarg = 0
            then \@hline
            else \@sline
         if
 if
END

\@sline ==
 BEGIN
   if \@xarg < 0
     then @negarg := T
          \@xarg  := -\@xarg
          \@yyarg := -\@yarg
     else @negarg := F
          \@yyarg := \@yarg
   fi
   \@tempcnta := |\@yyarg|
   if \@tempcnta > 6
     then error: 'LATEX ERROR: Illegal \line or \vector argument.'
          \@tempcnta := 0
   fi
   \box\@linechar := \hbox{\@linefnt \@getlinechar(\@xarg,\@yyarg) }
    if \@yarg > 0 then \@upordown = \raise
                         \@clnht := 0
                   else \@upordown = \lower
                        \@clnht := height of \box\@linechar
    fi
    \@clnwd  := width of \box\@linechar
    if @negarg
      then \hskip - width of \box\@linechar
           \reserved@a == \hskip - 2* width of box \@linechar
      else \reserved@a == \relax
    fi
 %% Put out integral number of line segments
```

```
    while \@clnwd <  \@linelen
      do  \@upordown \@clnht \copy\@linechar
          \reserved@a
          \@clnht := \@clnht + ht of \box\@linechar
          \@clnwd := \@clnwd + width of \box\@linechar
      od

 %% Put out last segment
    \@clnht := \@clnht - height of \box\@linechar
    \@clnwd := \@clnwd - width of \box\@linechar
    \@tempdima   := \@linelen - \@clnwd
    \@tempdimb   := \@tempdima - width of \box\@linechar
    if @negarg  then \hskip -\@tempdimb
                 else \hskip  \@tempdimb
    fi
    \@tempdima   := 1000 * \@tempdima
    \@tempcnta   := \@tempdima / width of \box\@linechar
    \@tempdima   := (\@tempcnta * ht of \box\@linechar)/1000
    \@clnht := \@clnht + \@tempdima
    if \@linelen < width of box\@linechar
        then \hskip width of box\@linechar
        else \hbox{\@upordown \@clnht \copy\@linechar}
    fi
END

\@hline ==
  BEGIN
    if \@xarg < 0 then  \hskip -\@linelen \fi
    \vrule height \@halfwidth depth \@halfwidth width \@linelen
    if \@xarg < 0 then  \hskip -\@linelen \fi
 END

\@vline == if \@yarg < 0 \@downline else \@upline  fi


\@getlinechar(X,Y) ==
  BEGIN
    \@tempcnta := 8*X - 9
    if Y > 0
      then \@tempcnta := \@tempcnta + Y
      else \@tempcnta := \@tempcnta - Y + 64
    fi
    \char\@tempcnta
  END

\vector(X,Y){LEN} ==
BEGIN
 \@xarg    := X
 \@yarg    := Y
 \@linelen := LEN * \unitlength
```

```
 if \@xarg = 0
    then \@vvector
    else if \@yarg = 0
            then \@hvector
            else \@svector
         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
```

End of historical LaTeX 2.09 comments.

\if@negarg

```
\newif\if@negarg
```

(*End definition for* `\if@negarg`*. This function is documented on page* **??**.)

\line

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\line}{default units}%
⟨latexrelease⟩\expandafter\let\csname line \endcsname\@undefind
\def\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@defaultunitsset\@linelen{#3}\unitlength
  \ifdim\@linelen<\z@\@badlinearg\else
    \ifnum\@xarg =\z@ \@vline
      \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
    \fi
  \fi}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\line}{default units}%
⟨latexrelease⟩\expandafter\let\csname line \endcsname\@undefind
⟨latexrelease⟩\def\line(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
⟨latexrelease⟩  \@linelen #3\unitlength
⟨latexrelease⟩  \ifdim\@linelen<\z@\@badlinearg\else
⟨latexrelease⟩    \ifnum\@xarg =\z@ \@vline
⟨latexrelease⟩      \else \ifnum\@yarg =\z@ \@hline \else \@sline\fi
⟨latexrelease⟩    \fi
```

```
⟨latexrelease⟩   \fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\line`. *This function is documented on page* ??.)

`\@sline`

```
\def\@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}
```

(*End definition for* `\@sline`. *This function is documented on page* **??**.)

`\@hline`

```
\def\@hline{\ifnum \@xarg <\z@ \hskip -\@linelen \fi
\vrule \@height \@halfwidth \@depth \@halfwidth \@width \@linelen
\ifnum \@xarg <\z@ \hskip -\@linelen \fi}
```

(*End definition for* `\@hline`. *This function is documented on page* **??**.)

`\@getlinechar`

```
\def\@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}
```

(*End definition for* `\@getlinechar`. *This function is documented on page* **??**.)

`\vector`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\vector}{default units}%
⟨latexrelease⟩\expandafter\let\csname vector \endcsname\@undefind
\def\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
  \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
  \ifnum\@tempcnta<5\relax
  \@defaultunitsset\@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}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\vector}{default units}%
⟨latexrelease⟩\expandafter\let\csname vector \endcsname\@undefind
⟨latexrelease⟩\def\vector(#1,#2)#3{\@xarg #1\relax \@yarg #2\relax
⟨latexrelease⟩  \@tempcnta \ifnum\@xarg<\z@ -\@xarg\else\@xarg\fi
⟨latexrelease⟩  \ifnum\@tempcnta<5\relax
⟨latexrelease⟩  \@linelen #3\unitlength
⟨latexrelease⟩  \ifdim\@linelen<\z@\@badlinearg\else
⟨latexrelease⟩    \ifnum\@xarg =\z@ \@vvector
⟨latexrelease⟩      \else \ifnum\@yarg =\z@ \@hvector \else \@svector\fi
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \else\@badlinearg\fi}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\vector`. *This function is documented on page* **??**.)

\@hvector

```
\def\@hvector{\@hline\hb@xt@\z@{\@linefnt
 \ifnum \@xarg <\z@ \@getlarrow(1,0)\hss\else
    \hss\@getrarrow(1,0)\fi}}
```

(*End definition for* `\@hvector`*. This function is documented on page* ??.)

\@vvector

```
\def\@vvector{\ifnum \@yarg <\z@ \@downvector \else \@upvector \fi}
```

(*End definition for* `\@vvector`*. This function is documented on page* ??.)

\@svector

```
\def\@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}
```

(*End definition for* `\@svector`*. This function is documented on page* ??.)

\@getlarrow

```
\def\@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}
```

(*End definition for* `\@getlarrow`*. This function is documented on page* ??.)

\@getrarrow

```
\def\@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}
```

(*End definition for* `\@getrarrow`*. This function is documented on page* ??.)

\@vline

```
\def\@vline{\ifnum \@yarg <\z@ \@downline \else \@upline\fi}
```

(*End definition for* `\@vline`*. This function is documented on page* **??**.)

`\@upline`

```
\def\@upline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \@linelen \@depth \z@\hss}}
```

(*End definition for* `\@upline`*. This function is documented on page* **??**.)

`\@downline`

```
\def\@downline{%
  \hb@xt@\z@{\hskip -\@halfwidth \vrule \@width \@wholewidth
   \@height \z@ \@depth \@linelen \hss}}
```

(*End definition for* `\@downline`*. This function is documented on page* **??**.)

`\@upvector`

```
\def\@upvector{\@upline\setbox\@tempboxa\hbox{\@linefnt\char 54}% '66
  \raise \@linelen \hb@xt@\z@{\lower \ht\@tempboxa\box\@tempboxa\hss}}
```

(*End definition for* `\@upvector`*. This function is documented on page* **??**.)

`\@downvector`

```
\def\@downvector{\@downline\lower \@linelen
      \hb@xt@\z@{\@linefnt\char 63  % '77
      \hss}}
```

(*End definition for* `\@downvector`*. This function is documented on page* **??**.)

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

```
\dashbox{D}(X,Y) ==
  BEGIN
  leave vertical mode
  \hb@xt@ 0pt {
        \baselineskip := 0pt
        \lineskip     := 0pt
  %% HORIZONTAL DASHES
        \@dashdim := X * \unitlength
        \@dashcnt := \@dashdim + 200 % to prevent roundoff error
        \@dashdim := D * \unitlength
        \@dashcnt := \@dashcnt / \@dashdim
        if \@dashcnt is odd
          then \@dashdim := 0pt
               \@dashcnt := (\@dashcnt + 1) / 2
          else \@dashdim := \@dashdim / 2
               \@dashcnt := \@dashcnt / 2 - 1
               \box\@dashbox   := \hbox{\vrule height \@halfwidth
                                depth \@halfwidth width \@dashdim}
               \put(0,0){\copy\@dashbox}
               \put(0,Y){\copy\@dashbox}
               \put(X,0){\hskip -\@dashdim\copy\@dashbox}
               \put(X,Y){\hskip -\@dashdim\box\@dashbox}
               \@dashdim := 3 * \@dashdim
        fi
```

```
\box\@dashbox := \hbox{\vrule height \@halfwidth
                   depth \@halfwidth width D * \unitlength
                   \hskip D * \unitlength}
\@tempcnta := 0
\put(0,0){\hskip \@dashdim
         while \@tempcnta < \@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
```

End of historical LaTeX 2.09 comments.

\dashbox

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\dashbox}{default units}%
⟨latexrelease⟩\expandafter\let\csname dashbox \endcsname\@undefind
\def\dashbox#1(#2,#3){\leavevmode\hb@xt@\z@{\baselineskip \z@skip
\lineskip \z@skip
\@defaultunitsset\@dashdim{#2}\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@defaultunitsset\@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{%
  \@defaultunitsset\@tempdimc{#1}\unitlength
  \vrule \@height \@halfwidth \@depth \@halfwidth \@width \@tempdimc
  \hskip\@tempdimc}%
\@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 }}%
\@defaultunitsset\@dashdim{#3}\unitlength
\@dashcnt \@dashdim \advance\@dashcnt 200
\@defaultunitsset\@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
\@defaultunitsset\@tempdimb{#1}\unitlength
\setbox\@dashbox\hbox{%
  \vrule \@width \@wholewidth \@height\@tempdimb}%
\@tempcnta\z@
\put(0,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
\do{\vskip\@tempdimb\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}\@tempcnta\z@
\put(#2,0){\hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
\do{\vskip\@tempdimb\copy\@dashbox\advance\@tempcnta \@ne }%
\vskip\@dashdim}}}\@makepicbox(#2,#3)}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\dashbox}{default units}%
⟨latexrelease⟩\expandafter\let\csname dashbox \endcsname\@undefind
⟨latexrelease⟩\def\dashbox#1(#2,#3){%
⟨latexrelease⟩\leavevmode\hb@xt@\z@{\baselineskip \z@skip
⟨latexrelease⟩\lineskip \z@skip
⟨latexrelease⟩\@dashdim #2\unitlength
⟨latexrelease⟩\@dashcnt \@dashdim \advance\@dashcnt 200
⟨latexrelease⟩\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
⟨latexrelease⟩\ifodd\@dashcnt\@dashdim \z@
⟨latexrelease⟩\advance\@dashcnt \@ne \divide\@dashcnt \tw@
⟨latexrelease⟩\else \divide\@dashdim \tw@ \divide\@dashcnt \tw@
⟨latexrelease⟩\advance\@dashcnt \m@ne
⟨latexrelease⟩\setbox\@dashbox \hbox{%
⟨latexrelease⟩  \vrule \@height \@halfwidth \@depth \@halfwidth
⟨latexrelease⟩  \@width \@dashdim}\put(0,0){\copy\@dashbox}%
⟨latexrelease⟩\put(0,#3){\copy\@dashbox}%
⟨latexrelease⟩\put(#2,0){\hskip-\@dashdim\copy\@dashbox}%
⟨latexrelease⟩\put(#2,#3){\hskip-\@dashdim\box\@dashbox}%
⟨latexrelease⟩\multiply\@dashdim \thr@@
⟨latexrelease⟩\fi
⟨latexrelease⟩\setbox\@dashbox \hbox{%
⟨latexrelease⟩  \vrule \@height \@halfwidth \@depth \@halfwidth
⟨latexrelease⟩  \@width #1\unitlength\hskip #1\unitlength}\@tempcnta\z@
⟨latexrelease⟩\put(0,0){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
⟨latexrelease⟩\do{\copy\@dashbox\advance\@tempcnta \@ne }}\@tempcnta\z@
⟨latexrelease⟩\put(0,#3){\hskip\@dashdim \@whilenum \@tempcnta <\@dashcnt
⟨latexrelease⟩\do{\copy\@dashbox\advance\@tempcnta \@ne }}%
⟨latexrelease⟩\@dashdim #3\unitlength
⟨latexrelease⟩\@dashcnt \@dashdim \advance\@dashcnt 200
⟨latexrelease⟩\@dashdim #1\unitlength\divide\@dashcnt \@dashdim
⟨latexrelease⟩\ifodd\@dashcnt \@dashdim \z@
```

```
⟨latexrelease⟩\advance\@dashcnt \@ne \divide\@dashcnt \tw@
⟨latexrelease⟩\else
⟨latexrelease⟩\divide\@dashdim \tw@ \divide\@dashcnt \tw@
⟨latexrelease⟩\advance\@dashcnt \m@ne
⟨latexrelease⟩\setbox\@dashbox\hbox{\hskip -\@halfwidth
⟨latexrelease⟩\vrule \@width \@wholewidth
⟨latexrelease⟩\@height \@dashdim}\put(0,0){\copy\@dashbox}%
⟨latexrelease⟩\put(#2,0){\copy\@dashbox}%
⟨latexrelease⟩\put(0,#3){\lower\@dashdim\copy\@dashbox}%
⟨latexrelease⟩\put(#2,#3){\lower\@dashdim\copy\@dashbox}%
⟨latexrelease⟩\multiply\@dashdim \thr@@
⟨latexrelease⟩\fi
⟨latexrelease⟩\setbox\@dashbox\hbox{\vrule \@width \@wholewidth
⟨latexrelease⟩\@height #1\unitlength}\@tempcnta\z@
⟨latexrelease⟩\put(0,0){%
⟨latexrelease⟩  \hskip -\@halfwidth \vbox{\@whilenum \@tempcnta <\@dashcnt
⟨latexrelease⟩  \do{\vskip #1\unitlength\copy\@dashbox
⟨latexrelease⟩      \advance\@tempcnta\@ne }%
⟨latexrelease⟩  \vskip\@dashdim}}\@tempcnta\z@
⟨latexrelease⟩\put(#2,0){%
⟨latexrelease⟩  \hskip -\@halfwidth \vbox{\@whilenum \@tempcnta<\@dashcnt
⟨latexrelease⟩  \do{\vskip #1\unitlength\copy\@dashbox
⟨latexrelease⟩      \advance\@tempcnta \@ne }%
⟨latexrelease⟩  \vskip\@dashdim}}}\@makepicbox(#2,#3)}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\dashbox`. *This function is documented on page* **??**.)

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

CIRCLES AND OVALS

USER COMMANDS:

`\circle{D}` : Produces the circle with the diameter as close as possible to D * `\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
```

End of historical LaTeX 2.09 comments.

`\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
```

(*End definition for* `\if@ovt` *and others. These functions are documented on page* **??**.)

`\@ovxx` `\@ovyy` `\@ovdx` `\@ovdy` `\@ovro` `\@ovri`

```
\newdimen\@ovxx
\newdimen\@ovyy
\newdimen\@ovdx
\newdimen\@ovdy
\newdimen\@ovro
\newdimen\@ovri
```

(*End definition for* `\@ovxx` *and others. These functions are documented on page* **??**.)

`\advance\@tempdima 2pt\relax` added 1 Nov 88 to fix bug in which size of drawn circle not monotonic function of argument of `\circle`, caused by different rounding for dimensions of large and small circles.

`\@getcirc`

```
\def\@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}
```

(*End definition for* `\@getcirc`*. This function is documented on page* **??**.)

`\@picture@warn`

Generic warning for lines, vectors (used in `\@sline`) and oval or circle (used in `\@getcirc`) are not available at right size.

```
\def\@picture@warn{\@latex@warning{%
     \string\oval, \string\circle, or \string\line\space
     size unavailable}}
```

(*End definition for* `\@picture@warn`*. This function is documented on page* **??**.)

`\@put`

```
\def\@put#1#2#3{\raise #2\hb@xt@\z@{\hskip #1#3\hss}}
```

(*End definition for* `\@put`*. This function is documented on page* **??**.)

`\oval`

```
\def\oval(#1,#2){\@ifnextchar[{\@oval(#1,#2)}{\@oval(#1,#2)[]}}
```

(*End definition for* `\oval`*. This function is documented on page* **??**.)

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2016/03/31}%
⟨latexrelease⟩                  {\@ovhlinetrue}%
⟨latexrelease⟩                  {Avoid almost zero length leaders}%
⟨*2ekernel | latexrelease⟩
```

`\if@ovvline` `\if@ovhline` Tests whether horizontal or vertical lines are needed.

```
\newif\if@ovvline \@ovvlinetrue
\newif\if@ovhline \@ovhlinetrue
%     \begin{macrocode}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@ovhlinetrue}%
⟨latexrelease⟩                  {Avoid almost zero length leaders}%
⟨latexrelease⟩\let\if@ovvline\@undefined
⟨latexrelease⟩\let\if@ovhline\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\if@ovvline` *and* `\if@ovhline`*. These functions are documented on page* **??**.)

`\@oval`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@oval}{default units}%
\def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue

  \@ovvlinefalse \@ovhlinefalse

  \@tfor\reserved@a :=#3\do{%
    \csname @ov\reserved@a false\endcsname}%
  \@defaultunitsset\@ovxx{#1}\unitlength
  \@defaultunitsset\@ovyy{#2}\unitlength

  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
  \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue \fi\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@
```

```
  \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
  \ifdim \@ovdy >\z@ \@ovvlinetrue \fi
  \@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}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2016/03/31}%
⟨latexrelease⟩                  {\@oval}{default units}%
⟨latexrelease⟩\def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
⟨latexrelease⟩  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
⟨latexrelease⟩  \@ovvlinefalse \@ovhlinefalse
⟨latexrelease⟩  \@tfor\reserved@a :=#3\do{%
⟨latexrelease⟩    \csname @ov\reserved@a false\endcsname}%
⟨latexrelease⟩  \@ovxx #1\unitlength
⟨latexrelease⟩  \@ovyy #2\unitlength
⟨latexrelease⟩  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx \@ovvlinetrue
⟨latexrelease⟩  \else \@ovyy \ifdim \@ovyy =\@ovxx \else \@ovhlinetrue
⟨latexrelease⟩   \fi\fi
⟨latexrelease⟩  \advance \@tempdimb -2\p@
⟨latexrelease⟩  \@getcirc \@tempdimb
⟨latexrelease⟩  \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
⟨latexrelease⟩  \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
⟨latexrelease⟩  \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
⟨latexrelease⟩  \ifdim \@ovdx >\z@ \@ovhlinetrue \fi
⟨latexrelease⟩  \ifdim \@ovdy >\z@ \@ovvlinetrue \fi
⟨latexrelease⟩  \@circlefnt \setbox\@tempboxa
⟨latexrelease⟩  \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
⟨latexrelease⟩  \if@ovl
⟨latexrelease⟩   \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@ovt \@ovhorz \kern -\@ovxx \fi
⟨latexrelease⟩  \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
⟨latexrelease⟩  \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
⟨latexrelease⟩  \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
⟨latexrelease⟩  \endgroup}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@oval}{default units}%
⟨latexrelease⟩\def\@oval(#1,#2)[#3]{\begingroup\boxmaxdepth \maxdimen
⟨latexrelease⟩  \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
⟨latexrelease⟩  \@tfor\reserved@a :=#3\do
⟨latexrelease⟩                 {\csname @ov\reserved@a false\endcsname}%
⟨latexrelease⟩  \@ovxx #1\unitlength
⟨latexrelease⟩  \@ovyy #2\unitlength
⟨latexrelease⟩  \@tempdimb \ifdim \@ovyy >\@ovxx \@ovxx\else \@ovyy \fi
⟨latexrelease⟩  \advance \@tempdimb -2\p@
⟨latexrelease⟩  \@getcirc \@tempdimb
```

```
⟨latexrelease⟩  \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
⟨latexrelease⟩  \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
⟨latexrelease⟩  \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
⟨latexrelease⟩  \@circlefnt \setbox\@tempboxa
⟨latexrelease⟩  \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
⟨latexrelease⟩  \if@ovl
⟨latexrelease⟩   \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@ovt \@ovhorz \kern -\@ovxx \fi
⟨latexrelease⟩  \if@ovb \raise \@ovyy \@ovhorz \fi}\advance\@ovdx\@ovro
⟨latexrelease⟩  \advance\@ovdy\@ovro \ht\@tempboxa\z@ \dp\@tempboxa\z@
⟨latexrelease⟩  \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
⟨latexrelease⟩  \endgroup}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@oval`*. This function is documented on page* **??**.)

`\@ovvert`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2016/03/31}%
⟨latexrelease⟩                 {\@ovvert}{Avoid almost zero length leaders}%
⟨*2ekernel | latexrelease⟩
\def\@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
    \if@ovvline \leaders\vrule \@width \@wholewidth \fi
    \vfil \nointerlineskip
    \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
      \hbox{\char \@tempcntb}%
    \else \kern \@ovdy \kern \@ovro \fi}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@ovvert}{Avoid almost zero length leaders}%
⟨latexrelease⟩\def\@ovvert#1#2{\vbox to\@ovyy{%
⟨latexrelease⟩    \if@ovb \@tempcntb \@tempcnta \advance \@tempcntb #1\relax
⟨latexrelease⟩      \kern -\@ovro \hbox{\char \@tempcntb}\nointerlineskip
⟨latexrelease⟩    \else \kern \@ovri \kern \@ovdy \fi
⟨latexrelease⟩    \leaders\vrule \@width \@wholewidth\vfil \nointerlineskip
⟨latexrelease⟩    \if@ovt \@tempcntb \@tempcnta \advance \@tempcntb #2\relax
⟨latexrelease⟩      \hbox{\char \@tempcntb}%
⟨latexrelease⟩    \else \kern \@ovdy \kern \@ovro \fi}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@ovvert`*. This function is documented on page* **??**.)

`\@ovhorz`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2016/03/31}%
⟨latexrelease⟩                 {\@ovhorz}{Avoid almost zero length leaders}%
```

```
⟨*2ekernel | latexrelease⟩
\def\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
    \if@ovr \else \kern \@ovdx \fi
    \if@ovhline \leaders \hrule \@height \@wholewidth \fi
    \hfil
    \if@ovl \else \kern \@ovdx \fi
    \kern \@ovri}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@ovhorz}{Avoid almost zero length leaders}%
⟨latexrelease⟩\def\@ovhorz{\hb@xt@\@ovxx{\kern \@ovro
⟨latexrelease⟩    \if@ovr \else \kern \@ovdx \fi
⟨latexrelease⟩    \leaders \hrule \@height \@wholewidth \hfil
⟨latexrelease⟩    \if@ovl \else \kern \@ovdx \fi
⟨latexrelease⟩    \kern \@ovri}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@ovhorz`*. This function is documented on page* **??**.)

`\circle`

```
\def\circle{\@inmatherr\circle\@ifstar\@dot\@circle}
```

(*End definition for* `\circle`*. This function is documented on page* **??**.)

`\@circle`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\@circle}{default units}%
\def\@circle#1{%
  \begingroup \boxmaxdepth \maxdimen
   \@defaultunitsset\@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}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@circle}{default units}%
⟨latexrelease⟩\def\@circle#1{%
⟨latexrelease⟩  \begingroup \boxmaxdepth \maxdimen \@tempdimb #1\unitlength
⟨latexrelease⟩   \ifdim \@tempdimb >15.5\p@ \@getcirc\@tempdimb
⟨latexrelease⟩      \@ovro\ht\@tempboxa
```

```
⟨latexrelease⟩      \setbox\@tempboxa\hbox{\@circlefnt
⟨latexrelease⟩       \advance\@tempcnta\tw@ \char \@tempcnta
⟨latexrelease⟩       \advance\@tempcnta\m@ne \char \@tempcnta
⟨latexrelease⟩       \kern -2\@tempdima
⟨latexrelease⟩       \advance\@tempcnta\tw@
⟨latexrelease⟩       \raise \@tempdima \hbox{\char\@tempcnta}%
⟨latexrelease⟩       \raise \@tempdima
⟨latexrelease⟩         \box\@tempboxa}\ht\@tempboxa\z@ \dp\@tempboxa\z@
⟨latexrelease⟩       \@put{-\@ovro}{-\@ovro}{\box\@tempboxa}%
⟨latexrelease⟩   \else  \@circ\@tempdimb{96}\fi\endgroup}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@circle`*. This function is documented on page* **??**.)

`\@dot` Internal form of `\circle*`.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\@dot}{default units}%
\def\@dot#1{%
  \@defaultunitsset\@tempdimb{#1}\unitlength
  \@circ\@tempdimb{112}}
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@dot}{default units}%
⟨latexrelease⟩\def\@dot#1{\@tempdimb #1\unitlength \@circ\@tempdimb{112}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@dot`*. This function is documented on page* **??**.)

`\@circ`

```
\def\@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}
```

(*End definition for* `\@circ`*. This function is documented on page* **??**.)

`\@xarg` `\@yarg` `\@yyarg` Counters used for manipulating the 'slope' arguments.

```
\newcount\@xarg
\newcount\@yarg
\newcount\@yyarg
```

(*End definition for* `\@xarg`*,* `\@yarg`*, and* `\@yyarg`*. These functions are documented on page* **??**.)

`\@multicnt` Counter used in `\multiput`, and also `\multicolumn`.

```
\newcount\@multicnt
```

(*End definition for* `\@multicnt`*. This function is documented on page* **??**.)

\@xdim Length registers.
\@ydim

```
\newdimen\@xdim
\newdimen\@ydim
```

(*End definition for* `\@xdim` *and* `\@ydim`*. These functions are documented on page* ??.)

\@linechar Box for holding a line segment character, for sloping lines.

```
\newbox\@linechar
```

(*End definition for* `\@linechar`*. This function is documented on page* ??.)

\@linelen Length of the line currently being built.

```
\newdimen\@linelen
```

(*End definition for* `\@linelen`*. This function is documented on page* ??.)

\@clnwd Height and width of current line segment.
\@clnht

```
\newdimen\@clnwd
\newdimen\@clnht
```

(*End definition for* `\@clnwd` *and* `\@clnht`*. These functions are documented on page* ??.)

\@dashdim `\dashbox` internal registers.
\@dashbox
\@dashcnt

```
\newdimen\@dashdim
\newbox\@dashbox
\newcount\@dashcnt
```

(*End definition for* `\@dashdim`*,* `\@dashbox`*, and* `\@dashcnt`*. These functions are documented on page* ??.)

Initialization: "`\thinlines`"

```
\let\@linefnt\tenln
\let\@circlefnt\tencirc
\@wholewidth\fontdimen8\tenln
\@halfwidth .5\@wholewidth
```

1.1 Curves

The new `\qbezier` command, based on the old `\bezier` defined in `bezier.sty`.
Historical LaTeX 2.09 comments (not necessarily accurate any more):

```
\qbezier[N] == \bezier{N}

\bezier{N}(AX,AY)(BX,BY)(CX,CY) ==
  BEGIN
    IF N = 0
       THEN \@xdima := |BX - AX|
            \@xb := |CX - BX|
            \@xa := Max(\@xa, \@xb)
            \@ya := |BY - AY|
            \@yb := |CY - BY|
            \@ya := Max(\@ya, \@yb)
            @sc := Max(\@xa, \@ya)
            %% The coefficient .5 below is the degree of overlap of
```

```
        %% successive points, where 1 is no overlap and 0 is
        %% complete overlap.  A coefficient of C multiplies
        %% the number of points plotted by 1/C.
        %%
        \@xa := .5 * \@halfwidth
        @sc := @sc / \@halfwidth
        @sc := Max(@sc, qbeziermax)
    ELSE @sc := N
@scp := @sc+1
\@xb := 2 * (BX - AX) * \unitlength
\@xa := ((CX-AX)*\unitlength - \@xb)/@sc
\@yb := 2 * (BY - AY) * \unitlength
\@ya := ((CY-AY)*\unitlength - \@yb)/@sc
\@pictdot := square rule of width \@wholewidth
\count@ := 0
WHILE \count@ < @scp
  DO  \@xdim := ((\count@*\@xa + @xb) / @sc) * \count@
      \@ydim := ((\count@*\@ya + @yb) / @sc) * \count@
      plot pt with relative coords (\@xdim,\@ydim)
      \count@ := \count@+1
  OD
```

End of historical LaTeX 2.09 comments.

`\qbeziermax` The maximum number of points to plot.

```
\def\qbeziermax{500}
```

(*End definition for* `\qbeziermax`*. This function is documented on page* **??**.)

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}
```

(*End definition for* `\qbezier`*. This function is documented on page* **??**.)

`\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 `(`.

```
\def\bezier#1)#2(#3)#4({\@bezier#1)(#3)(}
```

\@bezier

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@bezier}{default units}%
\def\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
  \ifnum #1=\z@
      \@defaultunitsset\@ovxx{#4}\unitlength
        \@defaultunitsset{\advance\@ovxx}{-#2}\unitlength
        \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
      \@defaultunitsset\@ovdx{#6}\unitlength
        \@defaultunitsset{\advance\@ovdx}{-#4}\unitlength
        \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
        \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
      \@defaultunitsset\@ovyy{#5}\unitlength
        \@defaultunitsset{\advance\@ovyy}{-#3}\unitlength
        \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
      \@defaultunitsset\@ovdy{#7}\unitlength
        \@defaultunitsset{\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
  \@defaultunitsset\@ovdx{#4}\unitlength
  \@defaultunitsset{\advance\@ovdx}{-#2}\unitlength
      \multiply\@ovdx \tw@
  \@defaultunitsset\@ovxx{#6}\unitlength
  \@defaultunitsset{\advance\@ovxx}{-#2}\unitlength
      \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
  \@defaultunitsset\@ovdy{#5}\unitlength
  \@defaultunitsset{\advance\@ovdy}{-#3}\unitlength
       \multiply\@ovdy \tw@
  \@defaultunitsset\@ovyy{#7}\unitlength
  \@defaultunitsset{\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 | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@bezier}{default units}%
⟨latexrelease⟩\def\@bezier#1(#2,#3)(#4,#5)(#6,#7){%
⟨latexrelease⟩  \ifnum #1=\z@
⟨latexrelease⟩      \@ovxx #4\unitlength
⟨latexrelease⟩        \advance\@ovxx -#2\unitlength
⟨latexrelease⟩        \ifdim \@ovxx<\z@ \@ovxx -\@ovxx \fi
⟨latexrelease⟩      \@ovdx #6\unitlength
⟨latexrelease⟩        \advance\@ovdx -#4\unitlength
⟨latexrelease⟩        \ifdim \@ovdx<\z@ \@ovdx -\@ovdx \fi
⟨latexrelease⟩        \ifdim \@ovxx<\@ovdx \@ovxx \@ovdx \fi
⟨latexrelease⟩      \@ovyy #5\unitlength
⟨latexrelease⟩        \advance\@ovyy -#3\unitlength
⟨latexrelease⟩        \ifdim \@ovyy<\z@ \@ovyy -\@ovyy \fi
⟨latexrelease⟩      \@ovdy #7\unitlength
⟨latexrelease⟩        \advance\@ovdy -#5\unitlength
⟨latexrelease⟩        \ifdim \@ovdy<\z@  \@ovdy -\@ovdy \fi
⟨latexrelease⟩        \ifdim \@ovyy<\@ovdy \@ovyy  \@ovdy \fi
⟨latexrelease⟩      \@multicnt
⟨latexrelease⟩         \ifdim \@ovxx>\@ovyy \@ovxx \else \@ovyy \fi
⟨latexrelease⟩      \@ovxx .5\@halfwidth \divide\@multicnt\@ovxx
⟨latexrelease⟩      \ifnum
⟨latexrelease⟩        \qbeziermax<\@multicnt \@multicnt\qbeziermax\relax
⟨latexrelease⟩      \fi
⟨latexrelease⟩  \else \@multicnt#1\relax \fi
⟨latexrelease⟩  \@tempcnta\@multicnt \advance\@tempcnta\@ne
⟨latexrelease⟩  \@ovdx #4\unitlength \advance\@ovdx -#2\unitlength
⟨latexrelease⟩      \multiply\@ovdx \tw@
⟨latexrelease⟩  \@ovxx #6\unitlength \advance\@ovxx -#2\unitlength
⟨latexrelease⟩      \advance\@ovxx -\@ovdx \divide\@ovxx\@multicnt
⟨latexrelease⟩  \@ovdy #5\unitlength \advance\@ovdy -#3\unitlength
⟨latexrelease⟩       \multiply\@ovdy \tw@
⟨latexrelease⟩  \@ovyy #7\unitlength \advance\@ovyy -#3\unitlength
⟨latexrelease⟩      \advance\@ovyy -\@ovdy \divide\@ovyy\@multicnt
⟨latexrelease⟩  \setbox\@tempboxa\hbox{%
⟨latexrelease⟩            \hskip -\@halfwidth
⟨latexrelease⟩            \vrule \@height\@halfwidth
⟨latexrelease⟩                   \@depth \@halfwidth
⟨latexrelease⟩                   \@width \@wholewidth}%
⟨latexrelease⟩   \put(#2,#3){%
⟨latexrelease⟩     \count@\z@
⟨latexrelease⟩     \@whilenum{\count@<\@tempcnta}\do
⟨latexrelease⟩        {\@xdim\count@\@ovxx
⟨latexrelease⟩           \advance\@xdim\@ovdx
```

```
⟨latexrelease⟩            \divide\@xdim\@multicnt
⟨latexrelease⟩            \multiply\@xdim\count@
⟨latexrelease⟩          \@ydim\count@\@ovyy
⟨latexrelease⟩             \advance\@ydim\@ovdy
⟨latexrelease⟩             \divide\@ydim\@multicnt
⟨latexrelease⟩             \multiply\@ydim\count@
⟨latexrelease⟩          \raise \@ydim
⟨latexrelease⟩             \hb@xt@\z@{\kern\@xdim
⟨latexrelease⟩                          \unhcopy\@tempboxa\hss}%
⟨latexrelease⟩          \advance\count@\@ne}}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\bezier` *and* `\@bezier`*. These functions are documented on page* **??**.)

As the commands above all use "picture" interface we couldn't define them with `\DeclareRobustCommand` so we do that now.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                 {\bezier}{Make commands robust}%
\MakeRobust\bezier
\MakeRobust\circle
\MakeRobust\dashbox
\MakeRobust\line
\MakeRobust\linethickness
\MakeRobust\multiput
\MakeRobust\oval
\MakeRobust\put
\MakeRobust\qbezier
\MakeRobust\shortstack
\MakeRobust\thinlines
\MakeRobust\vector
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\bezier}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\bezier
⟨latexrelease⟩\kernel@make@fragile\circle
⟨latexrelease⟩\kernel@make@fragile\dashbox
⟨latexrelease⟩\kernel@make@fragile\line
⟨latexrelease⟩\kernel@make@fragile\linethickness
⟨latexrelease⟩\kernel@make@fragile\multiput
⟨latexrelease⟩\kernel@make@fragile\oval
⟨latexrelease⟩\kernel@make@fragile\put
⟨latexrelease⟩\kernel@make@fragile\qbezier
⟨latexrelease⟩\kernel@make@fragile\shortstack
⟨latexrelease⟩\kernel@make@fragile\thinlines
⟨latexrelease⟩\kernel@make@fragile\vector
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
⟨/2ekernel⟩
```

File J
ltthm.dtx

1 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*⟩.
Historical LaTeX 2.09 comments (not necessarily accurate any more):

```
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 nonexistent
        then ERROR
        else
             if \NAME is definable
               then BEGIN
                    \theNAME == \theOLDNAME
                    \NAME == \@thm{OLDNAME}{TEXT}
                    \endNAME == \@endtheorem
                    END
               else  error
             fi
      fi
    END

  \@thm{NAME}{TEXT} ==
    BEGIN
     \refstepcounter{NAME}
     if next char = [
        then \@ythm{NAME}{TEXT}
        else \@xthm{NAME}{TEXT}
     fi
    END

  \@xthm{NAME}{TEXT} ==
    BEGIN
     \@begintheorem{TEXT}{\theNAME}
     \ignorespaces
    END

  \@ythm{NAME}{TEXT}[OPARG] ==
    BEGIN
     \@opargbegintheorem{TEXT}{\theNAME}{OPARG}
     \ignorespaces
    END
```

End of historical LaTeX 2.09 comments.

\newtheorem

\newtheorem ought really be allowed only in the preamble Which would be good document style, and allow some main memory to be saved by declaring these commands to be \@onlypreamble. Unfortunately the LaTeX book indicates that \newtheorem may be used anywhere in the document...

```
⟨*2ekernel⟩
\def\newtheorem#1{%
  \@ifnextchar[{\@othm{#1}}{\@nthm{#1}}}
```

(*End definition for* \newtheorem*. This function is documented on page* ??.)

\@nthm

```
\def\@nthm#1#2{%
  \@ifnextchar[{\@xnthm{#1}{#2}}{\@ynthm{#1}{#2}}}
```

(*End definition for* \@nthm*. This function is documented on page* ??.)

\@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}}}
```

(*End definition for* \@xnthm*. This function is documented on page* ??.)

\@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}}}
```

(*End definition for* \@ynthm*. This function is documented on page* ??.)

\@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}}}}
```

(*End definition for* \@othm*. This function is documented on page* ??.)

\@thm

```
\def\@thm#1#2{%
  \refstepcounter{#1}%
  \@ifnextchar[{\@ythm{#1}{#2}}{\@xthm{#1}{#2}}}
```

(*End definition for* \@thm*. This function is documented on page* ??.)

\@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}
```

(*End definition for* `\@xthm` *and* `\@ythm`*. These functions are documented on page* **??**.)

Default values

\@thmcounter
\@thmcountersep

```
\def\@thmcounter#1{\noexpand\arabic{#1}}
\def\@thmcountersep{.}
```

(*End definition for* `\@thmcounter` *and* `\@thmcountersep`*. These functions are documented on page* **??**.)

\@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⟩
```

(*End definition for* `\@begintheorem`*,* `\@opargbegintheorem`*, and* `\@endtheorem`*. These functions are documented on page* **??**.)

File K
ltsect.dtx

1 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,}
```

1.1 The Title

`\title` `\author` `\date` `\thanks` `\and` `\maketitle`

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.

And finally, the `\maketitle` command produces the actual title, using the information previously saved with the other commands.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\title}{Make commands robust}%
```

`\title`

`\title` for use in `\maketitle`. If not given `\maketitle` will produce an error message.

```
\DeclareRobustCommand\title[1]{\gdef\@title{#1}}
```

(*End definition for* `\title`*. This function is documented on page* **??**.)

`\author`

`\author` for use in `\maketitle`. If not given `\maketitle` will produce a warning message.

```
\DeclareRobustCommand*\author[1]{\gdef\@author{#1}}
```

(*End definition for* `\author`*. This function is documented on page* **??**.)

`\date`

`\date` for use in `\maketitle`. If not given `\maketitle` will produce `\today` as the default.

```
\DeclareRobustCommand*\date[1]{\gdef\@date{#1}}
```

(*End definition for* `\date`*. This function is documented on page* **??**.)

`\thanks`

```
\DeclareRobustCommand\thanks[1]{\footnotemark
    \protected@xdef\@thanks{\@thanks
        \protect\footnotetext[\the\c@footnote]{#1}}%
}
```

(*End definition for* `\thanks`*. This function is documented on page* **??**.)

\and

```
\DeclareRobustCommand\and{%   % \begin{tabular}
  \end{tabular}%
  \hskip 1em \@plus.17fil%
  \begin{tabular}[t]{c}}%     % \end{tabular}
```

(*End definition for* \and*. This function is documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\title}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\title
⟨latexrelease⟩\kernel@make@fragile\author
⟨latexrelease⟩\kernel@make@fragile\date
⟨latexrelease⟩\kernel@make@fragile\thanks
⟨latexrelease⟩\kernel@make@fragile\and
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

\@title

```
\def\@title{\@latex@error{No \noexpand\title given}\@ehc}
```

(*End definition for* \@title*. This function is documented on page* ??.)

\@author

```
\def\@author{\@latex@warning@no@line{No \noexpand\author given}}
```

(*End definition for* \@author*. This function is documented on page* ??.)

\@date

```
\gdef\@date{\today}
```

(*End definition for* \@date*. This function is documented on page* ??.)

\@thanks

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

(*End definition for* \@thanks*. This function is documented on page* ??.)

```
\message{sectioning,}
```

1.2 Sectioning

\@secpenalty

```
\newcount\@secpenalty
\@secpenalty = -300
```

(*End definition for* \@secpenalty*. This function is documented on page* ??.)

\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
```

(*End definition for* `\if@noskipsec` *and* `\@noskipsectrue`*. These functions are documented on page* **??**.)

`\@startsection` The `\@startsection{`⟨*name*⟩`}{`⟨*level*⟩`}{ `⟨*indent*⟩`}{`⟨*beforeskip*⟩`}{`⟨*afterskip*⟩`}{`⟨*style*⟩`}*[`⟨*altheading*⟩ `]{`⟨*heading*⟩`}` command is the mother of all the user level sectioning commands. The part after the `*`, including the `*` is optional.

name: e.g., 'subsection'

level: a number, denoting depth of section – e.g., chapter = 0, section = 1, etc.

indent: Indentation of heading from left margin

beforeskip: Absolute value = skip to leave above the heading. If negative, then paragraph indent of text following heading is suppressed.

afterskip: if positive, then skip to leave below heading, else negative of skip to leave to right of run-in heading.

style: Commands to set style. Since June 1996 release the *last* command in this argument may be a command such as `\MakeUppercase` or `\fbox` that takes an argument. The section heading will be supplied as the argument to this command. So setting `#6` to, say, `\bfseries\MakeUppercase` would produce bold, uppercase headings.

If '`*`' is missing, then increment the counter. If it is present, then there should be no `[`⟨*altheading*⟩`]` argument. The command uses the counter 'secnumdepth'. It contains a pointer to the highest section level that is to be numbered.

Warning: The `\@startsection` command should be at the same or higher grouping level as the text that follows it. For example, you should *not* do something like

```
\def\foo{ \begingroup ...
            \paragraph{...}
          \endgroup}
```

Pseudocode for the `\@startsection` command *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
 \@startsection
{NAME}{LEVEL}{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE} ==
    BEGIN
     IF  @noskipsec = T  THEN  \leavevmode  FI
                               % true if previous section had no body.

     \par
     \@tempskipa  := BEFORESKIP
     @afterindent := T
     IF \@tempskipa < 0  THEN  \@tempskipa  := -\@tempskipa
                               @afterindent := F
     FI
     IF @nobreak = true
       THEN \everypar == null
       ELSE \addpenalty{\@secpenalty}
            \addvspace{\@tempskipa}
     FI
     IF * next
```

```
        THEN \@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}
        ELSE \@dblarg{\@sect
                 {NAME}{LEVEL}{INDENT}
                 {BEFORESKIP}{AFTERSKIP}{STYLE}}
     FI
 END
```

End of historical LaTeX 2.09 comments.

```
\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}}}}
```

(*End definition for* `\@startsection`*. This function is documented on page* **??**.)

`\@sect` Pseudocode for the `\@sect` command *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
 \@sect{NAME}{LEVEL}
        {INDENT}{BEFORESKIP}{AFTERSKIP}
        {STYLE}[ARG1]{ARG2}
            ==
   BEGIN
    IF LEVEL > \c@secnumdepth
      THEN \@svsec :=L null
      ELSE \refstepcounter{NAME}
           \@svsec :=L BEGIN \@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
```

End of historical LaTeX 2.09 comments.

```
\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}}
```

(*End definition for* `\@sect`*. This function is documented on page* **??**.)

`\@xsect` Pseudocode for the `\@xsect` command *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
\@xsect{AFTERSKIP} ==
 BEGIN
   IF AFTERSKIP > 0
     THEN \par \nobreak
          \vskip AFTERSKIP
          \@afterheading
     ELSE @nobreak :=G F
          @noskipsec :=G T
          \everypar{ IF @noskipsec = T
                       THEN @noskipsec :=G F
                            \clubpenalty := 10000 % local
                            \hskip -\parindent
                            \begingroup
                              \@svsechd
                            \endgroup
                            \unskip
                            \hskip -AFTERSKIP \relax
                                          %% relax added 14 Jan 91
                       ELSE \clubpenalty := \@clubpenalty % local
                            \everypar := NULL
                     FI
                   }
   FI

  END
```

End of historical LaTeX 2.09 comments.

```
\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}
```

(*End definition for* `\@xsect`*. This function is documented on page* **??**.)

`\@seccntformat` This command formats the section number including the space following it.

```
\def\@seccntformat#1{\csname the#1\endcsname\quad}
```

(*End definition for* `\@seccntformat`*. This function is documented on page* **??**.)

Pseudocode for the `\@ssect` command *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
\@ssect{INDENT}{BEFORESKIP}{AFTERSKIP}{STYLE}{ARG} ==
  BEGIN
   IF AFTERSKIP > 0
     THEN \begingroup
            STYLE
            \@hangfrom{\hskip INDENT}
                      {\interlinepenalty 10000 ARG\par}
          \endgroup
     ELSE \@svsechd == BEGIN STYLE
                           \hskip INDENT
                           ARG
                       END
   FI
   \@xsect{AFTERSKIP}
  END
```

End of historical LaTeX 2.09 comments.

Pseudocode for the `\@afterheading` command *Historical LaTeX 2.09 comments (not necessarily accurate any more):*

```
\@afterheading ==
 BEGIN
   @nobreak :=G true
   \everypar := BEGIN IF @nobreak = T
                      THEN @nobreak  :=G false
                           \clubpenalty := 10000 % local
                           IF @afterindent = F
                             THEN remove \lastbox
                           FI
                      ELSE \clubpenalty := \@clubpenalty % local
                           \everypar := NULL
                   FI
              END
 END
```

End of historical LaTeX 2.09 comments.

\@ssect

```
\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}}
```

(*End definition for* \@ssect*. This function is documented on page* ??.)

\if@afterindent
\@afterindenttrue

```
\newif\if@afterindent \@afterindenttrue
```

(*End definition for* \if@afterindent *and* \@afterindenttrue*. These functions are documented on page* ??.)

\@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}}
```

(*End definition for* \@afterheading*. This function is documented on page* ??.)

\@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}
```

(*End definition for* \@hangfrom*. This function is documented on page* ??.)

\c@secnumdepth
\c@tocdepth

```
\newcount\c@secnumdepth
\newcount\c@tocdepth
```

(*End definition for* \c@secnumdepth *and* \c@tocdepth*. These functions are documented on page* ??.)

\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}}}
```

(*End definition for* \secdef*. This function is documented on page* ??.)

1.2.1 Initializations

\sectionmark
\subsectionmark
\subsubsectionmark
\paragraphmark
\subparagraphmark

```
\let\sectionmark\@gobble
\let\subsectionmark\@gobble
\let\subsubsectionmark\@gobble
\let\paragraphmark\@gobble
\let\subparagraphmark\@gobble
```

(*End definition for* \sectionmark *and others. These functions are documented on page* ??.)

```
\message{contents,}
```

1.3 Table of Contents etc.

1.3.1 Convention

\tf@⟨*foo*⟩ = file number for output for table foo. The file is opened only if @filesw = true.

1.3.2 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.

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

\@starttoc{EXT} ==

```
BEGIN
  \begingroup
     \makeatletter
     read file \jobname.EXT
     IF @filesw = true
       THEN  open \jobname.EXT as file \tf@EXT
     FI
     @nobreak :=G FALSE  %% added 24 May 89
  \endgroup
END
```

End of historical LaTeX 2.09 comments.

```
\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}
```

(*End definition for* `\@starttoc`*. This function is documented on page* **??**.)

`\addcontentsline` The `\addcontentsline{`⟨*table*⟩`}{`⟨*type*⟩`}{`⟨*entry*⟩`}` command allows the user to add his/her own entry to a table of contents, etc. The command adds the entry `\contentsline{`⟨*type*⟩`}{`⟨*entry*⟩`}{`⟨*page*⟩`}` to the `.`⟨*table*⟩ file.

This macro is implemented as an application of `\addtocontents`. Note that `\thepage` is not expandable during `\protected@write` therefore one gets the page number at the time of the `\shipout`.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\addcontentsline}{fourth argument}%
\def\addcontentsline#1#2#3{%
```

We add an empty brace pair at the end of `\contentsline` so that the number of argument is identical in documents with and without hyperref.

```
  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}{}%
                     \protected@file@percent}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\addcontentsline}{Mask line endings}%
⟨latexrelease⟩ \def\addcontentsline#1#2#3{%
⟨latexrelease⟩  \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}%
```

We add `\protected@file@percent` at the end which is turned inside `\@writefile` into a percent character to mask the newline after the closing argument brace.

```
⟨latexrelease⟩             \protected@file@percent}}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\addcontentsline}{Mask line endings}%
```

```
⟨latexrelease⟩\def\addcontentsline#1#2#3{%
⟨latexrelease⟩   \addtocontents{#1}{\protect\contentsline{#2}{#3}{\thepage}}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\addcontentsline`*. This function is documented on page* **??**.)

`\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}}}
```

(*End definition for* `\addtocontents`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\contentsline`*. This function is documented on page* **??**.)

`\@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

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2018/12/01}%
⟨latexrelease⟩                  {\@dottedtocline}{Prevent protrusion}%
\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%
```

We finish off by preventing any protrusion if that is enabled. If protrusion happens the number may shift to the right and as a result you may end up with an additional dot in the toc line in some situations.

```
                        \kern-\p@\kern\p@}%
     \par}%
  \fi}
```

(*End definition for* \@dottedtocline*. This function is documented on page* ??.)

\noprotrusion

This command, if placed directly to the right (or left) of a word, will prevent protrusion of that word into the margin. It is used in the toc entry lines as they shouldn't protrude. It is implemented as to kerns that cancel each other but being there hide the word so that protrusion is not added. Note that a zero kern or an empty box would not work as the protrusion mechanism will skip over those.

```
\DeclareRobustCommand\noprotrusion{\leavevmode\kern-\p@\kern\p@}
```

(*End definition for* \noprotrusion*. This function is documented on page* ??.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@dottedtocline}{Prevent protrusion}%
⟨latexrelease⟩\def\@dottedtocline#1#2#3#4#5{%
⟨latexrelease⟩  \ifnum #1>\c@tocdepth \else
⟨latexrelease⟩    \vskip \z@ \@plus.2\p@
⟨latexrelease⟩    {\leftskip #2\relax \rightskip \@tocrmarg \parfillskip -\rightskip
⟨latexrelease⟩     \parindent #2\relax\@afterindenttrue
⟨latexrelease⟩     \interlinepenalty\@M
```

```
⟨latexrelease⟩      \leavevmode
⟨latexrelease⟩      \@tempdima #3\relax
⟨latexrelease⟩      \advance\leftskip \@tempdima \null\nobreak\hskip -\leftskip
⟨latexrelease⟩      {#4}\nobreak
⟨latexrelease⟩      \leaders\hbox{$\m@th
⟨latexrelease⟩         \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
⟨latexrelease⟩         mu$}\hfill
⟨latexrelease⟩      \nobreak
⟨latexrelease⟩      \hb@xt@\@pnumwidth{\hfil\normalfont \normalcolor #5}%
⟨latexrelease⟩      \par}%
⟨latexrelease⟩   \fi}
⟨latexrelease⟩
⟨latexrelease⟩\let\noprotrusion\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

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*⟩ flush-left in a box of width `\@tempdima` (Before 25 Jan 88 change, it also added `\@tempdima` to the hanging indentation.)

```
⟨2ekernel⟩
\def\numberline#1{\hb@xt@\@tempdima{#1\hfil}}
⟨/2ekernel⟩
```

(*End definition for* `\numberline`*. This function is documented on page* **??**.)

File L
ltfloat.dtx

1 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.

1.1 Floating Environments

```
⟨*2ekernel⟩
\message{floats,}
```

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

Where floats may appear on a page, and how many may appear there are specified by the following float placement parameters. The numbers are named like counters so the user can set them with the ordinary counter-setting commands.

```
\c@topnumber       : Number of floats allowed at the top of a column.
\topfraction       : Fraction of column that can be devoted to floats.
\c@dbltopnumber, \dbltopfraction
                   : Same as above, but for double-column floats.
\c@bottomnumber, \bottomfraction
                   : Same as above for bottom of page.
\c@totalnumber     : Number of floats allowed in a single column,
                         including in-text floats.
\textfraction      :Minimum fraction of column that must contain text.
\floatpagefraction: Minimum fraction of page that must be taken
                         up by float page.
\dblfloatpagefraction
                   : Same as above, for double-column floats.
```

The document style must define the following.

```
\fps@TYPE    : The default placement specifier for floats of type
               TYPE.

\ftype@TYPE : The type number for floats of type TYPE.

\ext@TYPE    : The file extension indicating the file on which the
               contents list for float type TYPE is stored.
                 For example,  \ext@figure = 'lof'.

\fnum@TYPE   : A macro to generate the figure number for a caption.
               For example, \fnum@TYPE == Figure \thefigure.

\@makecaption{NUM}{TEXT} :
            A macro to make a caption, with NUM the value
            produced by \fnum@... and TEXT the text of the caption.
            It can assume it's in a \parbox of the appropriate width.

\@float{TYPE}[PLACEMENT] : This macro begins a float environment for a
    single-column float of type TYPE with PLACEMENT as the placement
    specifier.  The default value of PLACEMENT is defined by
    \fps@TYPE.  The environment is ended by \end@float.
    E.g., \figure == \@float{figure}, \endfigure == \end@float.

 \@float{TYPE}[PLACEMENT] ==
  BEGIN
    if hmode then \@bsphack
                  \@floatpenalty := -10002
            else \@floatpenalty := -10003
    fi
    \@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`.

End of historical LaTeX 2.09 comments.

`\@floatpenalty`

```
3 \newcount\@floatpenalty
```

(*End definition for* `\@floatpenalty`. *This function is documented on page* **??**.)

\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}}%
}
```

(*End definition for* `\caption`*. This function is documented on page* **??**.)

\@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}
```

(*End definition for* `\@caption`*. This function is documented on page* **??**.)

\@float
\@dblflset

```
\def\@float#1{%
  \@ifnextchar[%
    {\@xfloat{#1}}%
    {\edef\reserved@a{\noexpand\@xfloat{#1}[\csname fps@#1\endcsname]}%
     \reserved@a}}
```

(*End definition for* `\@float` *and* `\@dblflset`*. These functions are documented on page* **??**.)

\@dblfloat

```
\def\@dblfloat{%
  \if@twocolumn\let\reserved@a\@dbflt\else\let\reserved@a\@float\fi
  \reserved@a}
```

(*End definition for* `\@dblfloat`*. This function is documented on page* **??**.)

`\fps@dbl` Note that all double floats have default fps 'tp'.

(*End definition for* `\fps@dbl`*. This function is documented on page* **??**.)

`\@setfps` This sets the fps, dealing with error conditions by adding the default.

(*End definition for* `\@setfps`*. This function is documented on page* **??**.)

`\@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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\@xfloat}{Check float options}%
⟨*2ekernel | latexrelease⟩
\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
```

Start of changes, use a nested if structure, ending in an error.

```
          {%
           \if \reserved@a h%
             \ifodd \@tempcnta
             \else
               \advance \@tempcnta \@ne
             \fi
```

```
          \else\if \reserved@a t%
            \@setfpsbit \tw@
          \else\if \reserved@a b%
            \@setfpsbit 4%
          \else\if \reserved@a p%
            \@setfpsbit 8%
          \else\if \reserved@a !%
            \ifnum \@tempcnta>15
              \advance\@tempcnta -\sixt@@n\relax
            \fi
          \else
            \@latex@error{Unknown float option `\reserved@a'}%
            {Option `\reserved@a' ignored and `p' used.}%
            \@setfpsbit 8%
          \fi\fi\fi\fi\fi
          }%
```

End of changes

```
      \@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
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@xfloat}{Check float options}%
⟨latexrelease⟩\def\@xfloat #1[#2]{%
⟨latexrelease⟩  \@nodocument
⟨latexrelease⟩  \def \@captype {#1}%
⟨latexrelease⟩   \def \@fps {#2}%
⟨latexrelease⟩   \@onelevel@sanitize \@fps
⟨latexrelease⟩   \def \reserved@b {!}%
⟨latexrelease⟩   \ifx \reserved@b \@fps
⟨latexrelease⟩     \@fpsadddefault
⟨latexrelease⟩   \else
⟨latexrelease⟩     \ifx \@fps \@empty
⟨latexrelease⟩       \@fpsadddefault
```

```
⟨latexrelease⟩      \fi
⟨latexrelease⟩    \fi
⟨latexrelease⟩    \ifhmode
⟨latexrelease⟩      \@bsphack
⟨latexrelease⟩      \@floatpenalty -\@Mii
⟨latexrelease⟩    \else
⟨latexrelease⟩      \@floatpenalty-\@Miii
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \ifinner
⟨latexrelease⟩      \@parmoderr\@floatpenalty\z@
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@next\@currbox\@freelist
⟨latexrelease⟩      {%
⟨latexrelease⟩       \@tempcnta \sixt@@n
⟨latexrelease⟩       \expandafter \@tfor \expandafter \reserved@a
⟨latexrelease⟩         \expandafter :\expandafter =\@fps
⟨latexrelease⟩         \do
⟨latexrelease⟩          {%
⟨latexrelease⟩           \if \reserved@a h%
⟨latexrelease⟩             \ifodd \@tempcnta
⟨latexrelease⟩             \else
⟨latexrelease⟩               \advance \@tempcnta \@ne
⟨latexrelease⟩             \fi
⟨latexrelease⟩           \fi
⟨latexrelease⟩           \if \reserved@a t%
⟨latexrelease⟩             \@setfpsbit \tw@
⟨latexrelease⟩           \fi
⟨latexrelease⟩           \if \reserved@a b%
⟨latexrelease⟩             \@setfpsbit 4%
⟨latexrelease⟩           \fi
⟨latexrelease⟩           \if \reserved@a p%
⟨latexrelease⟩             \@setfpsbit 8%
⟨latexrelease⟩           \fi
⟨latexrelease⟩           \if \reserved@a !%
⟨latexrelease⟩             \ifnum \@tempcnta>15
⟨latexrelease⟩               \advance\@tempcnta -\sixt@@n\relax
⟨latexrelease⟩             \fi
⟨latexrelease⟩           \fi
⟨latexrelease⟩           }%
⟨latexrelease⟩       \@tempcntb \csname ftype@\@captype \endcsname
⟨latexrelease⟩       \multiply \@tempcntb \@xxxii
⟨latexrelease⟩       \advance \@tempcnta \@tempcntb
⟨latexrelease⟩       \global \count\@currbox \@tempcnta
⟨latexrelease⟩       }%
⟨latexrelease⟩    \@fltovf
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \global \setbox\@currbox
⟨latexrelease⟩    \color@vbox
⟨latexrelease⟩      \normalcolor
⟨latexrelease⟩      \vbox \bgroup
⟨latexrelease⟩        \hsize\columnwidth
⟨latexrelease⟩        \@parboxrestore
⟨latexrelease⟩        \@floatboxreset
⟨latexrelease⟩}%
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@xfloat`*. This function is documented on page* **??**.)

`\@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
}
```

(*End definition for* `\@floatboxreset`*. This function is documented on page* **??**.)

`\@setnobreak`

```
\def \@setnobreak{%
  \if@nobreak
    \let\outer@nobreak\@nobreaktrue
    \@nobreakfalse
  \fi
}
```

(*End definition for* `\@setnobreak`*. This function is documented on page* **??**.)

`\@setminipage`

```
\def \@setminipage{%
  \@minipagetrue
  \everypar{\@minipagefalse\everypar{}}%
}
```

(*End definition for* `\@setminipage`*. This function is documented on page* **??**.)

`\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 definition for* `\end@float`*. This function is documented on page* ??.)

`\end@dblfloat`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\end@dblfloat}{float order in 2-column}%
⟨*2ekernel | latexrelease⟩
\def\end@dblfloat{%
  \if@twocolumn
    \@endfloatbox
    \ifnum\@floatpenalty <\z@
      \@largefloatcheck
```

Force the depth of two column float boxes.

```
      \global\dp\@currbox1sp %
```

What follows is essentially `\end@float` without a starting `\@endfloatbox`.

```
      \@cons\@currlist\@currbox
      \ifnum\@floatpenalty <-\@Mii
        \penalty -\@Miv
        \@tempdima\prevdepth
        \vbox{}%
        \prevdepth\@tempdima
        \penalty\@floatpenalty
      \else
        \vadjust{\penalty -\@Miv \vbox{}\penalty\@floatpenalty}\@Esphack
      \fi
    \fi
  \else
    \end@float
  \fi
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\end@dblfloat}{float order in 2-column}%
⟨latexrelease⟩\def\end@dblfloat{%
⟨latexrelease⟩\if@twocolumn
⟨latexrelease⟩  \@endfloatbox
⟨latexrelease⟩  \ifnum\@floatpenalty <\z@
```

We make sure that we never exceed `\textheight`, otherwise float will never get typeset (91/03/15 FMi).

```
⟨latexrelease⟩    \@largefloatcheck
⟨latexrelease⟩    \@cons\@dbldeferlist\@currbox
⟨latexrelease⟩  \fi
```

RmS 92/03/18 changed \@esphack to \@Esphack.

```
⟨latexrelease⟩     \ifnum \@floatpenalty =-\@Mii \@Esphack\fi
⟨latexrelease⟩\else
⟨latexrelease⟩   \end@float
⟨latexrelease⟩\fi
⟨latexrelease⟩}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \end@dblfloat. *This function is documented on page* ??.)

\@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
}
```

(*End definition for* \@endfloatbox. *This function is documented on page* ??.)

\outer@nobreak

```
\let\outer@nobreak\@empty
```

(*End definition for* \outer@nobreak. *This function is documented on page* ??.)

\@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
}
```

(*End definition for* \@largefloatcheck. *This function is documented on page* ??.)

\@dbflt
\@xdblfloat

```
\def\@dbflt#1{\@ifnextchar[{\@xdblfloat{#1}}{\@xdblfloat{#1}[tp]}}
\def\@xdblfloat#1[#2]{%
  \@xfloat{#1}[#2]\hsize\textwidth\linewidth\textwidth}
```

(*End definition for* \@dbflt *and* \@xdblfloat. *These functions are documented on page* ??.)

Moved to ltoutput 93/12/16

```
%\newcount\c@topnumber
%\newcount\c@dbltopnumber
%\newcount\c@bottomnumber
%\newcount\c@totalnumber
```

`\@floatplacement` 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}
⟨/2ekernel⟩
```

(*End definition for* `\@floatplacement`*. This function is documented on page* ??.)

`\@dblfloatplacement` This should be called only within a group. Now changed to provide extra checks in `\@addtodblcol`, needed when processing a BANG float.

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩          {\@dblfloatplacement}{float order in 2-column}%
⟨*2ekernel | latexrelease⟩
```

When making two column float area, look for floats with 1sp depth.

```
\def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
   \global\@dbltoproom \dbltopfraction\@colht
   \@textmin \@colht
   \advance \@textmin -\@dbltoproom
   \@fpmin \dblfloatpagefraction\textheight
   \@fptop \@dblfptop
   \@fpsep \@dblfpsep
   \@fpbot \@dblfpbot
```

`\f@depth` is used in `\@testwrongwidth` to look for either column or dbl-column floats. A value of `1sp` signals the latter. Because of this setting here, `\@dblfloatplacment` needs to be called inside a group which is a questionable design.

```
   \def\f@depth{1sp}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩          {\@dblfloatplacement}{float order in 2-column}%
⟨latexrelease⟩\def \@dblfloatplacement {%
```

Textpage bit: global, but need not be.

```
⟨latexrelease⟩  \global \@dbltopnum \c@dbltopnumber
⟨latexrelease⟩  \global \@dbltoproom \dbltopfraction\@colht
```

This new bit uses `\@textmin` to locally store the amount of extra room in the column.

```
⟨latexrelease⟩  \@textmin \@colht
⟨latexrelease⟩  \advance \@textmin -\@dbltoproom
```

Floatpage bit: must be local.

```
⟨latexrelease⟩  \@fpmin \dblfloatpagefraction\textheight
⟨latexrelease⟩  \@fptop \@dblfptop
⟨latexrelease⟩  \@fpsep \@dblfpsep
⟨latexrelease⟩  \@fpbot \@dblfpbot
⟨latexrelease⟩}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@dblfloatplacement`. *This function is documented on page* **??**.)

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

```
MARGINAL NOTES:

Marginal notes use the same mechanism as floats to communicate
with the \output routine.  Marginal notes are distinguished from
floats by having a negative placement specification.  The command
\marginpar [LTEXT]{RTEXT} generates a marginal note in a parbox,
using LTEXT if it's on the left and RTEXT if it's on the right.
(Default is RTEXT = LTEXT.)  It uses the following parameters.

\marginparwidth : Width of marginal notes.
\marginparsep   : Distance between marginal note and text.
      the page layout to determine how to move the marginal
      note into the margin.   E.g., \@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
```

End of historical LaTeX 2.09 comments.

`\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}
```

(*End definition for* `\marginpar`*. This function is documented on page* **??**.)

`\@xmpar`

```
\long\def\@xmpar[#1]#2{%
  \@savemarbox\@marbox{#1}%
  \@savemarbox\@currbox{#2}%
  \@xympar}
```

(*End definition for* `\@xmpar`*. This function is documented on page* **??**.)

`\@ympar`

```
\long\def\@ympar#1{%
  \@savemarbox\@marbox{#1}%
  \global\setbox\@currbox\copy\@marbox
  \@xympar}
```

(*End definition for* `\@ympar`*. This function is documented on page* **??**.)

\@savemarbox

```
\long\def \@savemarbox #1#2{%
  \global\setbox #1%
    \color@vbox
      \vtop{%
        \hsize\marginparwidth
        \@parboxrestore
        \@marginparreset
        #2%
        \@minipagefalse
        \outer@nobreak
        }%
    \color@endbox
}
```

(*End definition for* `\@savemarbox`*. This function is documented on page* **??**.)

\@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
}
```

(*End definition for* `\@marginparreset`*. This function is documented on page* **??**.)

\@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
}
```

(*End definition for* `\@xympar`*. This function is documented on page* **??**.)

\reversemarginpar
\normalmarginpar

```
\def\reversemarginpar{\global\@mparbottom\z@ \@reversemargintrue}
\def\normalmarginpar{\global\@mparbottom\z@ \@reversemarginfalse}
```

(*End definition for* `\reversemarginpar` *and* `\normalmarginpar`*. These functions are documented on page* ??.)

```
\message{footnotes,}
```

1.2 Footnotes

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

`\footnote{NOTE}` : User command to insert a footnote.

`\footnote[NUM]{NOTE}`: User command to insert a footnote numbered NUM, where NUM is a number – 1, 2, etc. For example, if footnotes are numbered *, **, etc. within pages, then `\footnote[2]{...}` produces footnote '**'. This command does not step the footnote counter.

`\footnotemark[NUM]` : Command to produce just the footnote mark in the text, but no footnote. With no argument, it steps the footnote counter before generating the mark.

`\footnotetext[NUM]{TEXT}` : Command to produce the footnote but no mark. `\footnote` is equivalent to `\footnotemark \footnotetext` .

As in PLAIN, footnotes use `\insert\footins,` and the following parameters:

`\footnotesize` : Size-changing command for footnotes.

`\footnotesep` : The height of a strut placed at the beginning of every footnote.

`\skip\footins` : Space between main text and footnotes. The rule separating footnotes from text occurs in this space. This space lies above the strut of height `\footnotesep` which is at the beginning of the first footnote.

`\footnoterule` : Macro to draw the rule separating footnotes from text. It is executed right after a `\vspace` of `\skip\footins.` It should take zero vertical space–i.e., it should to a negative skip to compensate for any positive space it occupies. (See PLAIN.TEX.)

`\interfootnotelinepenalty` : Interline penalty for footnotes.

`\thefootnote` : In usual LaTeX style, produces the footnote number. If footnotes are to be numbered within pages, then the document style file must include an `\@addtoreset`

command to cause the footnote counter to be reset when the page counter is stepped. This is not a good idea, though, because the counter will not always be reset in time to ensure that the first footnote on a page is footnote number one.

`\@thefnmark` : Holds the current footnote's mark–e.g., `\dag` or '1' or 'a'.

`\@mpfnnumber` : A macro that generates the numbers for `\footnote` and `\footnotemark` commands. It == `\thefootnote` outside a minipage environment, but can be changed inside to generate numbers for `\footnote`'s.

`\@makefnmark` : A macro to generate the footnote marker from `\@thefnmark` The default definition was `\hbox{$^\@thefnmark$}`.

This is now replaced by `\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
```

End of historical LaTeX 2.09 comments.

\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
```

(*End definition for* \footins*. This function is documented on 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
```

(*End definition for* \footnoterule*. This function is documented on page* ??.)

\thefootnote

```
\@definecounter{footnote}
\def\thefootnote{\@arabic\c@footnote}
```

(*End definition for* \thefootnote*. This function is documented on page* ??.)

\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}}
```

(*End definition for* \thempfootnote*. This function is documented on page* ??.)

\@makefnmark Default definition.

```
%\def\@makefnmark{\hbox{$^{\@thefnmark}\m@th$}}
\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
```

(*End definition for* \@makefnmark*. This function is documented on page* ??.)

\textsuperscript This command provides superscript characters in the current text font. It's implementation might change!!!

```
\DeclareRobustCommand*\textsuperscript[1]{%
  \@textsuperscript{\selectfont#1}}
```

(*End definition for* \textsuperscript*. This function is documented on page* ??.)

\@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.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                    {\@textsuperscript}{superscript baseline}%
\def\@textsuperscript#1{%
  {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\sf@size#1}}}}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@textsuperscript}{superscript baseline}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@textsuperscript#1{%
⟨latexrelease⟩  {\m@th\ensuremath{^{\mbox{\fontsize\sf@size\z@#1}}}}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@textsuperscript`*. This function is documented on page* **??**.)

`\textsubscript`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\textsubscript}{\textsubscript}%
⟨*2ekernel | latexrelease⟩
\DeclareRobustCommand*\textsubscript[1]{%
  \@textsubscript{\selectfont#1}}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\textsubscript}{\textsubscript}%
⟨latexrelease⟩\let\textsubscript\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\textsubscript`*. This function is documented on page* **??**.)

`\@textsubscript`

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@textsubscript}{subscript baseline}%
\def\@textsubscript#1{%
  {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\sf@size#1}}}}}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\@textsubscript}{subscript baseline}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@textsubscript#1{%
⟨latexrelease⟩  {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@textsubscript}{subscript baseline}%
⟨latexrelease⟩\let\@textsubscript\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@textsubscript`*. This function is documented on page* **??**.)

`\footnotesep`

```
\newdimen\footnotesep
```

(*End definition for* `\footnotesep`*. This function is documented on page* **??**.)

`\footnote`

```
\def\footnote{\@ifnextchar[\@xfootnote{\stepcounter\@mpfn
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark\@footnotetext}}
```

(*End definition for* `\footnote`*. This function is documented on page* **??**.)

`\@xfootnote`

```
\def\@xfootnote[#1]{%
   \begingroup
     \csname c@\@mpfn\endcsname #1\relax
     \unrestored@protected@xdef\@thefnmark{\thempfn}%
   \endgroup
   \@footnotemark\@footnotetext}
```

(*End definition for* `\@xfootnote`*. This function is documented on page* **??**.)

`\@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}}%
```

(*End definition for* `\@footnotetext`*. This function is documented on page* **??**.)

`\footnotemark`

```
\def\footnotemark{%
   \@ifnextchar[\@xfootnotemark
     {\stepcounter{footnote}%
      \protected@xdef\@thefnmark{\thefootnote}%
      \@footnotemark}}
```

(*End definition for* `\footnotemark`*. This function is documented on page* **??**.)

`\@xfootnotemark`

```
\def\@xfootnotemark[#1]{%
   \begingroup
      \c@footnote #1\relax
      \unrestored@protected@xdef\@thefnmark{\thefootnote}%
   \endgroup
   \@footnotemark}
```

(*End definition for* `\@xfootnotemark`*. This function is documented on page* **??**.)

\@footnotemark

```
\def\@footnotemark{%
  \leavevmode
  \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
  \@makefnmark
  \ifhmode\spacefactor\@x@sf\fi
  \relax}
```

(*End definition for* `\@footnotemark`*. This function is documented on page* **??**.)

\footnotetext

```
\def\footnotetext{%
     \@ifnextchar [\@xfootnotenext
       {\protected@xdef\@thefnmark{\thempfn}%
    \@footnotetext}}
```

(*End definition for* `\footnotetext`*. This function is documented on page* **??**.)

\@xfootnotenext

```
\def\@xfootnotenext[#1]{%
  \begingroup
     \csname c@\@mpfn\endcsname #1\relax
     \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotetext}
```

(*End definition for* `\@xfootnotenext`*. This function is documented on page* **??**.)

\thempfn
\@mpfn

```
\def\@mpfn{footnote}
\def\thempfn{\thefootnote}
⟨/2ekernel⟩
```

(*End definition for* `\thempfn` *and* `\@mpfn`*. These functions are documented on page* **??**.)

File M
ltidxglo.dtx

1 Index and Glossary Generation

Index and Glossary commands.

`\makeindex` A preamble command to turn on indexing.

`\makeglossary` A preamble command to turn on making glossary entries.

`\index` Make an index entry for `#1`.

`\glossary` Make a glossary entry for `#1`.

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

```
\makeindex ==
  BEGIN
            \index ==  BEGIN \@bsphack
                             \begingroup
                                \protect{X} == \string X\space
                                 %% added 3 Feb 87 for \index commands
                                 %% in \footnotes
                                 re-\catcode special characters
                                 to 'other'
                                 \@wrindex
  END

\@wrindex{ITEM} ==
  BEGIN
      write of {\indexentry{ITEM}{page number}}
    \endgroup
    \@esphack
  END

INITIALIZATION:

\index == BEGIN \@bsphack
                \begingroup
                   re-\catcode special characters (in case '%' there)
                   \@index
          END

\@index{ITEM} == BEGIN \endgroup \@esphack END
```

Changes made 14 Apr 89 to write `\glossaryentry`'s instead of `\indexentry`'s on the .glo file.

End of historical LaTeX 2.09 comments.

```
⟨*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
```

(*End definition for* `\makeindex`*. This function is documented on page* **??**.)

\@wrindex

```
\def\@wrindex#1{%
   \protected@write\@indexfile{}%
      {\string\indexentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

(*End definition for* `\@wrindex`*. This function is documented on page* **??**.)

\index

```
\def\index{\@bsphack\begingroup \@sanitize\@index}
```

(*End definition for* `\index`*. This function is documented on page* **??**.)

\@index

```
\def\@index#1{\endgroup\@esphack}
```

(*End definition for* `\@index`*. This function is documented on page* **??**.)

\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
```

(*End definition for* `\makeglossary`*. This function is documented on page* **??**.)

\@wrglossary

```
\def\@wrglossary#1{%
   \protected@write\@glossaryfile{}%
      {\string\glossaryentry{#1}{\thepage}}%
 \endgroup
 \@esphack}
```

(*End definition for* `\@wrglossary`*. This function is documented on page* **??**.)

\glossary

```
\def\glossary{\@bsphack\begingroup\@sanitize\@index}
```

(*End definition for* `\glossary`*. This function is documented on page* **??**.)

⟨/2ekernel⟩

File N
ltbibl.dtx

1 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,}
```

Historical LATEX 2.09 comments (not necessarily accurate any more):

PARAMETERS

```
\@cite    : A macro such that \@cite{LABEL1,LABEL2}{NOTE}
            produces the output for a \cite[NOTE]{FOO1,FOO2} command,
            where entry FOOi is defined by \bibitem[LABELi]{FOOi}.
            The switch @tempswa is true if the optional NOTE argument
            is present.
            The default definition is :
              \@cite{LABELS}{NOTE} ==
                 BEGIN [LABELS
                        IF @tempswa = T THEN , NOTE FI
                        ]
                 END

\@biblabel : A macro to produce the label in the bibliography
             entry.  For \bibitem[LABEL]{NAME}, the label is
```

generated by \@biblabel{LABEL}. It has the default definition \@biblabel{LABEL} -> [LABEL].

CONVENTION

\b@FOO : The name or number of the reference created by \cite{FOO} E.g., if \cite{FOO} -> [17] , then \b@FOO -> 17.

End of historical LaTeX 2.09 comments.

\bibitem

```
\def\bibitem{\@ifnextchar[\@lbibitem\@bibitem}
```

(*End definition for* \bibitem*. This function is documented on page* ??.)

\@lbibitem

```
\def\@lbibitem[#1]#2{\item[\@biblabel{#1}\hfill]\if@filesw
      {\let\protect\noexpand
       \immediate
       \write\@auxout{\string\bibcite{#2}{#1}}}\fi\ignorespaces}
```

(*End definition for* \@lbibitem*. This function is documented on page* ??.)

\@bibitem

```
\def\@bibitem#1{\item\if@filesw \immediate\write\@auxout
       {\string\bibcite{#1}{\the\value{\@listctr}}}\fi\ignorespaces}
```

(*End definition for* \@bibitem*. This function is documented on page* ??.)

\bibcite

```
\def\bibcite{\@newl@bel b}
```

(*End definition for* \bibcite*. This function is documented on page* ??.)

\citation

```
\let\citation\@gobble
```

(*End definition for* \citation*. This function is documented on page* ??.)

\cite

```
\DeclareRobustCommand\cite{%
  \@ifnextchar [{\@tempswatrue\@citex}{\@tempswafalse\@citex[]}}
```

(*End definition for* \cite*. This function is documented on page* ??.)

\@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}}
```

(*End definition for* \@citex*. This function is documented on page* ??.)

\bibdata
\bibstyle

```
\let\bibdata=\@gobble
\let\bibstyle=\@gobble
```

(*End definition for* \bibdata *and* \bibstyle*. These functions are documented on page* ??.)

\bibliography

```
\def\bibliography#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibdata{\zap@space#1 \@empty}}%
  \fi
  \@input@{\jobname.bbl}}
```

(*End definition for* \bibliography*. This function is documented on page* ??.)

\bibliographystyle

```
\def\bibliographystyle#1{%
  \ifx\@begindocumenthook\@undefined\else
    \expandafter\AtBeginDocument
  \fi
    {\if@filesw
       \immediate\write\@auxout{\string\bibstyle{#1}}%
     \fi}}
```

(*End definition for* \bibliographystyle*. This function is documented on page* ??.)

\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
```

(*End definition for* `\nocite`*. This function is documented on page* **??**.)

1.1 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}]}
```

(*End definition for* `\@cite`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@cite@ofmt`*. This function is documented on page* **??**.)

`\@biblabel`

```
\def\@biblabel#1{[#1]}
⟨/2ekernel⟩
```

(*End definition for* `\@biblabel`*. This function is documented on page* **??**.)

File O
ltpage.dtx

1 Page styles and related commands

1.1 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.

1.2 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.)

1.3 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}}}
```

(*End definition for* `\pagestyle`*. This function is documented on page* **??**.)

\thispagestyle User command to set the page style for this page only.

```
\def\thispagestyle#1{%
  \@ifundefined{ps@#1}%
    \undefinedpagestyle
    {\global\@specialpagetrue\gdef\@specialstyle{#1}}}
```

(*End definition for* `\thispagestyle`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\ps@empty`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\ps@plain`*. This function is documented on page* **??**.)

\@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
```

(*End definition for* `\@leftmark` *and* `\@rightmark`*. These functions are documented on page* **??**.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\markboth}{Make commands robust}%
```

\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`

```
\DeclareRobustCommand*\markboth[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}
```

```
\DeclareRobustCommand*\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}
```

(*End definition for* `\markboth` *and* `\markright`*. These functions are documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\markboth}{Make commands robust}%
⟨latexrelease⟩
⟨latexrelease⟩\kernel@make@fragile\markboth
⟨latexrelease⟩\kernel@make@fragile\markright
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\@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}
```

(*End definition for* `\@markright`*,* `\leftmark`*, and* `\rightmark`*. These functions are documented on page* **??**.)

`\@themark` Initialise LaTeX's marks without setting a TeX mark ⟨*whatsit*⟩.

```
\def\@themark{{}{}}
```

(*End definition for* `\@themark`*. This function is documented on page* **??**.)

`\mark` Test versions of LaTeX 2ε initialised TeX's `\mark` system at this point, but this was removed before the first release.

```
AtBeginDocument{\mark{{}{}}}
```

(*End definition for* `\mark`*. This function is documented on page* **??**.)

`\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`.)

```
\DeclareRobustCommand\raggedbottom{%
  \def\@textbottom{\vskip \z@ \@plus.0001fil}\let\@texttop\relax}
```

(*End definition for* `\raggedbottom`*. This function is documented on page* **??**.)

`\flushbottom` `\flushbottom`: Inverse of `\raggedbottom` — makes all pages the same height.

```
\DeclareRobustCommand\flushbottom{%
  \let\@textbottom\relax \let\@texttop\relax}
```

(*End definition for* `\flushbottom`*. This function is documented on page* **??**.)

`\sloppy` `\sloppy` will never (well, hardly ever) produce overfull boxes, but may produce underfull ones. (14 June 85)

```
\DeclareRobustCommand\sloppy{%
  \tolerance 9999%
  \emergencystretch 3em%
  \hfuzz .5\p@
  \vfuzz\hfuzz}
```

(*End definition for* `\sloppy`*. This function is documented on page* **??**.)

`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.

```
\DeclareRobustCommand\fussy{%
  \emergencystretch\z@
  \tolerance 200%
  \hfuzz .1\p@
  \vfuzz\hfuzz}
```

(*End definition for* `\fussy`*. This function is documented on page* **??**.)

`\overfullrule` LATEX default is no overfull box rule. Changed by document class option.

```
\overfullrule 0pt
```

(*End definition for* `\overfullrule`*. This function is documented on page* **??**.)

⟨/2ekernel⟩

File P
ltclass.dtx

1 Introduction

This file implements the following declarations, which replace `\documentstyle` in LaTeX 2_ε documents.

Note that old documents containing `\documentstyle` will be run using a compatibility option—thus keeping everyone happy, we hope!

The overall idea is that there are two types of 'style files': 'class files' which define elements and provide a default formatting for them; and 'packages' which provide extra functionality. One difference between LaTeX 2_ε and LaTeX2.09 is that LaTeX 2_ε packages may have options. Note that options to classes packages may be implemented such that they input files, but these file names are not necessarily directly related to the option name.

2 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 can also be called with an optional argument which is used to alter some of its behavior: option `force` or `overwrite` will allow for overwriting existing files, option `nosearch` will only check the current directory when looking if the file exists. This can be useful if you want to generate a local (modified) copy of some file that is already in the search tree of TEX. Finally, you can use `noheader` to prevent it from writing the standard blurb at the top of the file (this is actually the same as using the star form of the environment).

The environment is now allowed anywhere in the document, but to ensure that all packages or options necessary are available when the document is run, it is normally best to place it at the top of your file (before `\documentclass`). A possible use case for using it inside the document body is if you want to reuse some text several times in the document you could then write it and later use `\input` to retrieve it where needed.

The begin and end tags should each be on a line by itself.

2.1 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.

3 Class and Package interface

3.1 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.

3.2 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.

3.3 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}
```

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 is the analogous command for packages.

\RequirePackageWithOptions

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`

To find out if a package has already been loaded, use `\@ifpackageloaded{`⟨*package*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

`\@ifpackagelater` `\@ifclasslater`

To find out if a package has already been loaded with a version equal to or more recent than ⟨*version*⟩, use `\@ifpackagelater{`⟨*package*⟩`}{`⟨*version*⟩`}{`⟨*true*⟩`}{`⟨*false*⟩`}`.

`\@ifpackagewith` `\@ifclasswith`

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).

3.4 Declaring new options

Options for classes and packages are built using the same macros.

`\DeclareOption`

To define a builtin option, use `\DeclareOption{`⟨*name*⟩`}{`⟨*code*⟩`}`.

`\DeclareOption*`

To define the default action to perform for local options which have not been declared, use `\DeclareOption*{`⟨*code*⟩`}`.

Note: there should be no use of `\RequirePackage`, `\DeclareOption`, `\DeclareOption*` or `\ProcessOptions` inside `\DeclareOption` or `\DeclareOption*`.

Possible uses for `\DeclareOption*` include:

`\DeclareOption*{}`

Do nothing. Silently accept unknown options. (This suppresses the usual warnings.)

`\DeclareOption*{\@unkownoptionerror}`

Complain about unknown local options. (The initial setting for package files.)

`\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{`⟨*pkg-name*⟩`}}`

Handle the current option by passing it on to the package ⟨*pkg-name*⟩, which will presumably be loaded via `\RequirePackage` later in the file. This is useful for building 'extension' packages, that perhaps handle a couple of new options, but then pass everything else on to an existing package.

```
\DeclareOption*{\InputIfFileExists{xx-\CurrentOption.yyy}%
                {}%
                {\OptionNotUsed}}
```

Handle the option foo by loading the file `xx-foo.yyy` if it exists, otherwise do nothing, but declare that the option was not used. Actually the `\OptionNotUsed` declaration is only needed if this is being used in class files, but does no harm in package files.

3.5 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.

4 Implementation

```
⟨*2ekernel⟩
```

\if@compatibility The flag for compatibility mode.

```
\newif\if@compatibility
```

(*End definition for* `\if@compatibility`*. This function is documented on page* **??**.)

\@documentclasshook This legacy hook is called after the first `\documentclass` command. It is *not* integrated with the new 2020 hook management system! 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
}
```

(*End definition for* `\@documentclasshook`*. This function is documented on page* **??**.)

\@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
```

(*End definition for* `\@declaredoptions`*. This function is documented on page* **??**.)

\@classoptionslist List of options of the main class.

```
\let\@classoptionslist\relax
\@onlypreamble\@classoptionslist
```

(*End definition for* `\@classoptionslist`*. This function is documented on page* **??**.)

\@unusedoptionlist List of options of the main class that haven't been declared or loaded as class option files.

```
\let\@unusedoptionlist\@empty
\@onlypreamble\@unusedoptionlist
```

(*End definition for* \@unusedoptionlist. *This function is documented on page* ??.)

\CurrentOption Name of current package or option.

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

(*End definition for* \CurrentOption. *This function is documented on page* ??.)

\@currpath Path to the current file if explicitly given.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@currpath}%
⟨latexrelease⟩   {Add \@currpath}%
\let\@currpath\@empty
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@currpath}%
⟨latexrelease⟩   {Add \@currpath}%
⟨latexrelease⟩\let\@currpath\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
```

(*End definition for* \@currpath. *This function is documented on page* ??.)

\@currname Name of current package or option.

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

(*End definition for* \@currname. *This function is documented on page* ??.)

\@currext The current file extension.

```
\global\let\@currext=\@empty
```

(*End definition for* \@currext. *This function is documented on page* ??.)

\@clsextension
\@pkgextension The two possible values of \@currext.

```
\def\@clsextension{cls}
\def\@pkgextension{sty}
\@onlypreamble\@clsextension
\@onlypreamble\@pkgextension
```

(*End definition for* \@clsextension *and* \@pkgextension. *These functions are documented on page* ??.)

\@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.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩
```

```
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@pushfilename}%
⟨latexrelease⟩  {Add \@expl@push@filename@@ and \@expl@push@filename@aux@@}%
\def\@pushfilename{%
```

The push and pop macros are injected in `\@pushfilename` and `\@popfilename` so that they correctly keep track of the hook labels.

This needs cleanup with the expl3 interfaces also playing here, e.g., `\@expl@push@filename@@` needs cleanup and (and should probably not have this name either).

```
  \@expl@@@hook@curr@name@push@@n{}%
  \@expl@push@filename@@
  \xdef\@currnamestack{%
    {\@currname}%
    {\@currext}%
    {\the\catcode`\@}%
    \@currnamestack}%
```

Temporarily add a stack for `\@currpath` here. This should be integrated in the main file stack eventually, but other packages rely on `\@currnamestack` having three elements per file, so that isn't a trivial change. The prefix `\@kernel@...` hopefully discourages people from using it.

```
  \xdef\@kernel@currpathstack{%
    {\@currpath}%
    \@kernel@currpathstack}%
  \@expl@push@filename@aux@@}
⟨latexrelease⟩\EndIncludeInRelease
```

The following version of `\@pushfilename` didn't formally exist in this file, but in the 2020/02/02 release, expl3 was preloaded and it patched `\@pushfilename` (and `\@popfilename`) by adding some hooks in there. But rolling back to 2020/02/02, expl3 doesn't patch these macros again, so rolling back has to take those hooks into account. Same goes for `\@popfilename`.

```
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}{\@pushfilename}%
⟨latexrelease⟩  {Add \@expl@push@filename@@}%
⟨latexrelease⟩\def\@pushfilename{%
⟨latexrelease⟩  \@expl@push@filename@@
⟨latexrelease⟩  \xdef\@currnamestack{%
⟨latexrelease⟩    {\@currname}%
⟨latexrelease⟩    {\@currext}%
⟨latexrelease⟩    {\the\catcode`\@}%
⟨latexrelease⟩    \@currnamestack}%
⟨latexrelease⟩    \@expl@push@filename@aux@@}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@pushfilename}%
⟨latexrelease⟩  {Add \@expl@push@filename@@ and \@expl@push@filename@aux@@}%
⟨latexrelease⟩\def\@pushfilename{%
⟨latexrelease⟩  \xdef\@currnamestack{%
⟨latexrelease⟩    {\@currname}%
⟨latexrelease⟩    {\@currext}%
⟨latexrelease⟩    {\the\catcode`\@}%
⟨latexrelease⟩    \@currnamestack}}
⟨latexrelease⟩\EndIncludeInRelease
\@onlypreamble\@pushfilename
```

```
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@popfilename}%
⟨latexrelease⟩  {Add \@expl@pop@filename@@}%
\def\@popfilename{\@expl@@@hook@curr@name@pop@@
  \expandafter\@p@pfilename\@currnamestack\@nil
```

Same for popping:

```
  \expandafter\@p@pfilepath\@kernel@currpathstack\@nil
  \@expl@pop@filename@@}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/02/02}{\@popfilename}%
⟨latexrelease⟩  {Add \@expl@push@filename@@}%
⟨latexrelease⟩\def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil
⟨latexrelease⟩  \@expl@pop@filename@@}
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@popfilename}%
⟨latexrelease⟩  {Add \@expl@push@filename@@ and \@expl@push@filename@aux@@}%
⟨latexrelease⟩\def\@popfilename{\expandafter\@p@pfilename\@currnamestack\@nil}
⟨latexrelease⟩\EndIncludeInRelease
\@onlypreamble\@popfilename
```

```
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
```

```
\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
```

(*End definition for* `\@pushfilename`*,* `\@popfilename`*, and* `\@currnamestack`*. These functions are documented on page* ??.)

`\@kernel@currpathstack` Path to the current file if explicitly given. The auxiliary is needed here to insert a `\@empty` to prevent the loss of braces.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@kernel@currpathstack}%
⟨latexrelease⟩  {Add \@kernel@currpathstack}%
\gdef\@kernel@currpathstack{}
\def\@p@pfilepath#1{%
  \gdef\@currpath{#1}\@p@pfilepath@aux\@empty}
\def\@p@pfilepath@aux#1\@nil{%
  \xdef\@kernel@currpathstack{#1}}
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@kernel@currpathstack}%
⟨latexrelease⟩  {Add \@kernel@currpathstack}%
⟨latexrelease⟩\let\@kernel@currpathstack\@undefined
```

```
⟨latexrelease⟩\let\@p@pfilepath\@undefined
⟨latexrelease⟩\let\@p@pfilepath@aux\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
⟨*2ekernel⟩
```

(*End definition for* `\@kernel@currpathstack`. *This function is documented on page* **??**.)

`\@ptionlist` Returns the option list of the file.

```
\def\@ptionlist#1{%
  \@ifundefined{opt@#1}\@empty{\csname opt@#1\endcsname}}
\@onlypreamble\@ptionlist
```

(*End definition for* `\@ptionlist`. *This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@ifpackageloaded` *and* `\@ifclassloaded`. *These functions are documented on page* **??**.)

`\@ifpackagelater` `\@ifclasslater`

`\@ifpackagelater{`⟨*name*⟩`}{YYYY/MM/DD}{`⟨*true code*⟩`}{`⟨*false code*⟩`}` Checks that the package loaded is more recent or equal to the given date. A better name for it would therefore been `\@ifpackagelaterorequal` but it is in use for more than 30 years, so ...

```
\def\@ifpackagelater{\@ifl@ter\@pkgextension}
\def\@ifclasslater{\@ifl@ter\@clsextension}
\@onlypreamble\@ifpackagelater
\@onlypreamble\@ifclasslater
```

(*End definition for* `\@ifpackagelater` *and* `\@ifclasslater`. *These functions are documented on page* **??**.)

`\IfPackageAtLeastTF` `\IfClassAtLeastTF` `\IfFormatAtLeastTF`

`\IfFormatAtLeastTF{YYYY/MM/DD}{`⟨*true code*⟩`}{`⟨*false code*⟩`}` Test if the format is later or equal to the given date.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\IfFormatAtLeastTF}{Test format date}%
\def\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
\let\IfPackageAtLeastTF\@ifpackagelater
\let\IfClassAtLeastTF\@ifclasslater
\@onlypreamble\IfFormatAtLeastTF
\@onlypreamble\IfPackageAtLeastTF
\@onlypreamble\IfClassAtLeastTF
```

For rollback pretend it was available since the beginning of dawn.

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\IfFormatAtLeastTF}{Test format date}%
⟨latexrelease⟩\def\IfFormatAtLeastTF{\@ifl@t@r\fmtversion}
⟨latexrelease⟩\let\IfPackageAtLeastTF\@ifpackagelater
⟨latexrelease⟩\let\IfClassAtLeastTF\@ifclasslater
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\IfPackageAtLeastTF`*,* `\IfClassAtLeastTF`*, and* `\IfFormatAtLeastTF`*. These functions are documented on page* ??.)

`\@ifl@ter`

```
\def\@ifl@ter#1#2{%
  \expandafter\@ifl@t@r
    \csname ver@#2.#1\endcsname}
\@onlypreamble\@ifl@ter
⟨/2ekernel⟩
```

This internal macro is also used in `\NeedsTeXFormat`.

```
⟨latexrelease⟩\IncludeInRelease{2018/04/01}%
⟨latexrelease⟩                  {\@ifl@t@r}{Guard against bad input}%
⟨*2ekernel | latexrelease⟩
\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}
\def\@parse@version@#1{\@parse@version0#1}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@ifl@t@r}{Guard against bad input}%
⟨latexrelease⟩\def\@ifl@t@r#1#2{%
⟨latexrelease⟩  \ifnum\expandafter\@parse@version#1//00\@nil<%
⟨latexrelease⟩        \expandafter\@parse@version#2//00\@nil
⟨latexrelease⟩    \expandafter\@secondoftwo
⟨latexrelease⟩  \else
⟨latexrelease⟩    \expandafter\@firstoftwo
⟨latexrelease⟩  \fi}
⟨latexrelease⟩\let\@parse@version@\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\@ifl@t@r
```

(*End definition for* `\@ifl@ter`*. This function is documented on page* ??.)

```
⟨/2ekernel⟩
⟨*2ekernel | latexreleasefirst⟩
\def\@parse@version#1/#2/#3#4#5\@nil{%
\@parse@version@dash#1-#2-#3#4\@nil
}
```

The \if test here ensures that an argument with no / or - produces 0 (actually 00).

```
\def\@parse@version@dash#1-#2-#3#4#5\@nil{%
  \if\relax#2\relax\else#1\fi#2#3#4 }
⟨/2ekernel | latexreleasefirst⟩
⟨*2ekernel⟩
```

\@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 shouldn't use \CurrentOption here... (changed to \reserved@b.)

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease⟩                  {\@if@pti@ns}{Spaces in option clash check}%
⟨*2ekernel | latexrelease⟩
\def\@if@pti@ns#1#2{%
 \let\reserved@a\@firstoftwo
```

```
 \edef\reserved@b{\zap@space#2 \@empty}%
 \@for\reserved@b:=\reserved@b\do{%
   \ifx\reserved@b\@empty
   \else
     \expandafter\in@\expandafter{\expandafter,\reserved@b,}{,#1,}%
     \ifin@
     \else
       \let\reserved@a\@secondoftwo
     \fi
   \fi
 }%
 \reserved@a}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@if@pti@ns}{Spaces in option clash check}%
⟨latexrelease⟩\def\@if@pti@ns#1#2{%
⟨latexrelease⟩ \let\reserved@a\@firstoftwo
⟨latexrelease⟩ \@for\reserved@b:=#2\do{%
⟨latexrelease⟩  \ifx\reserved@b\@empty
⟨latexrelease⟩   \else
⟨latexrelease⟩   \expandafter\in@\expandafter
⟨latexrelease⟩                    {\expandafter,\reserved@b,}{,#1,}%
⟨latexrelease⟩    \ifin@
⟨latexrelease⟩    \else
⟨latexrelease⟩     \let\reserved@a\@secondoftwo
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩ }%
⟨latexrelease⟩ \reserved@a}
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@onlypreamble\@if@pti@ns
```

(*End definition for* `\@ifpackagewith` *and* `\@ifclasswith`*. These functions are documented on page* **??**.)

`\ProvidesPackage` Checks that the current filename is correct, and defines `\ver@filename`.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩  {\ProvidesPackage}{Check name with \strcmp}%
⟨*2ekernel | latexrelease⟩
\def\ProvidesPackage#1{%
  \xdef\@gtempa{#1}%
```

Here `\@currpath` is explicitly added to the file name to report when a package or class is loaded using an explicit path. Loading using a path in the argument is supported but not encouraged.

```
  \@expandtwoargs\@expl@str@if@eq@@nnTF
      {\@gtempa}{\@currpath\@currname}{}{%
    \@latex@warning@no@line{You have requested
      \@cls@pkg\space`\@currpath\@currname',\MessageBreak
       but the \@cls@pkg\space provides `#1'}%
    }%
  \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}}%]
\@onlypreamble\ProvidesPackage
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\ProvidesPackage}{Undo: check name with \strcmp}%
⟨latexrelease⟩\def\ProvidesPackage#1{%
⟨latexrelease⟩  \xdef\@gtempa{#1}%
⟨latexrelease⟩  \ifx\@gtempa\@currname\else
⟨latexrelease⟩    \@latex@warning@no@line{You have requested
⟨latexrelease⟩      \@cls@pkg\space`\@currname',\MessageBreak
⟨latexrelease⟩       but the \@cls@pkg\space provides `#1'}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}}%]
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\ProvidesPackage`*. This function is documented on page* **??**.)

`\@pr@videpackage` This is the helper command for `\ProvidesPackage`. It tries to be cautious when handling the identification string in case it contains UTF-8 characters.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\@pr@videpackage}{Allow for package subsitution}%
\def\@pr@videpackage[#1]{%
  \expandafter\protected@xdef                %     <-- protected...
     \csname ver@\@currname.\@currext\endcsname{#1}% Loaded package
  \expandafter\let
    \csname ver@\@currpkg@reqd\expandafter\endcsname % Requested package
```

```
    \csname ver@\@currname.\@currext\endcsname
  \ifx\@currext\@clsextension
    \typeout{Document Class: \@gtempa\space#1}%
  \else
    \protected@wlog{Package: \@gtempa\space#1}%   <--- protected
  \fi}
```

(*End definition for* `\@pr@videpackage`. *This function is documented on page* **??**.)

`\protected@wlog` This is like plain TEX's `\wlog` but gracefully handles protected commands.

```
\long\def\protected@wlog#1{\begingroup
  \set@display@protect
  \immediate \write \m@ne {#1}\endgroup }
```

(*End definition for* `\protected@wlog`. *This function is documented on page* **??**.)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2020/02/02}%
⟨latexrelease⟩                 {\@pr@videpackage}{Protection for package info}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@pr@videpackage[#1]{%
⟨latexrelease⟩  \expandafter\protected@xdef                %     <-- protected...
⟨latexrelease⟩     \csname ver@\@currname.\@currext\endcsname{#1}%
⟨latexrelease⟩\ifx\@currext\@clsextension
⟨latexrelease⟩    \typeout{Document Class: \@gtempa\space#1}%
⟨latexrelease⟩  \else
⟨latexrelease⟩    \protected@wlog{Package: \@gtempa\space#1}%   <--- protected
⟨latexrelease⟩  \fi}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@pr@videpackage}{Protection for package info}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@pr@videpackage[#1]{%
⟨latexrelease⟩  \expandafter\xdef\csname ver@\@currname.\@currext\endcsname{#1}%
⟨latexrelease⟩  \ifx\@currext\@clsextension
⟨latexrelease⟩    \typeout{Document Class: \@gtempa\space#1}%
⟨latexrelease⟩  \else
⟨latexrelease⟩    \wlog{Package: \@gtempa\space#1}%
⟨latexrelease⟩  \fi}
⟨latexrelease⟩\let\protected@wlog\@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\@pr@videpackage
```

`\ProvidesClass` Like `\ProvidesPackage`, but for classes. This needs a dummy latexrelease block to copy the definition of `\ProvidesPackage` as it changes across releases.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\ProvidesClass}{Track \ProvidesPackage}%
⟨*2ekernel | latexrelease⟩
\let\ProvidesClass\ProvidesPackage
\@onlypreamble\ProvidesClass
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\ProvidesClass`. *This function is documented on page* **??**.)

`\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 definition for* `\ProvidesFile` *and* `\@providesfile`. *These functions are documented on page* **??**.)

`\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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}{\@pass@ptions}
⟨latexrelease⟩  {Add file replacement in \@pass@ptions}%
⟨*2ekernel | latexrelease⟩
\def\@pass@ptions#1#2#3{%
  \@expl@@@filehook@set@curr@file@@nNN
    {\@expl@@@filehook@resolve@file@subst@@w #3.#1\@nil}%
      \reserved@a\reserved@b
  \@expl@@@filehook@clear@replacement@flag@@
  \expandafter\xdef\csname opt@\reserved@a\endcsname{%
    \@ifundefined{opt@\reserved@a}\@empty
      {\csname opt@\reserved@a\endcsname,}%
    \zap@space#2 \@empty}%

  \expandafter\let
    \csname opt@#3.#1\expandafter\endcsname
    \csname opt@\reserved@a\endcsname}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
%
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@pass@ptions}
```

```
⟨latexrelease⟩  {\@pass@ptions}%
⟨latexrelease⟩\def\@pass@ptions#1#2#3{%
⟨latexrelease⟩  \expandafter\xdef\csname opt@#3.#1\endcsname{%
⟨latexrelease⟩    \@ifundefined{opt@#3.#1}\@empty
⟨latexrelease⟩      {\csname opt@#3.#1\endcsname,}%
⟨latexrelease⟩    \zap@space#2 \@empty}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@onlypreamble\@pass@ptions
```

```
\def\PassOptionsToPackage{\@pass@ptions\@pkgextension}
\def\PassOptionsToClass{\@pass@ptions\@clsextension}
\@onlypreamble\PassOptionsToPackage
\@onlypreamble\PassOptionsToClass
```

(*End definition for* `\PassOptionsToPackage` *and* `\PassOptionsToClass`*. These functions are documented on page* ??.)

`\DeclareOption` `\DeclareOption*`

Adds an option as a `\ds@` command, or the default `\default@ds` command.

```
\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
```

(*End definition for* `\DeclareOption` *and* `\DeclareOption*`*. These functions are documented on page* ??.)

`\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
```

(*End definition for* `\OptionNotUsed`*. This function is documented on page* ??.)

`\default@ds`

The default option code. Set by `\@onefilewithoptions` to either `\OptionNotUsed` for classes, or `\@unknownoptionerror` for packages. This may be reset in either case with `\DeclareOption*`.

```
% \let\default@ds\OptionNotUsed
```

(*End definition for* `\default@ds`*. This function is documented on page* ??.)

\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*.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\@process@pti@ns}{Unused options issue}%
\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{\expandafter\let
                     \csname unprocessedoptions-\@currname.\@currext\endcsname
                     \relax}}
\@onlypreamble\@process@pti@ns
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\@process@pti@ns}{Unused options issue}%
⟨latexrelease⟩
⟨latexrelease⟩\def\@process@pti@ns{%
⟨latexrelease⟩  \@for\CurrentOption:=\@curroptions\do{%
⟨latexrelease⟩    \@ifundefined{ds@\CurrentOption}%
⟨latexrelease⟩      {\@use@ption
⟨latexrelease⟩       \default@ds}%
⟨latexrelease⟩      \@use@ption}%
⟨latexrelease⟩  \@for\CurrentOption:=\@declaredoptions\do{%
⟨latexrelease⟩    \expandafter\let\csname ds@\CurrentOption\endcsname\relax}%
⟨latexrelease⟩  \let\CurrentOption\@empty
⟨latexrelease⟩  \let\@fileswith@pti@ns\@@fileswith@pti@ns
⟨latexrelease⟩  \AtEndOfPackage{\let\@unprocessedoptions\relax}}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\ProcessOptions` *and* `\ProcessOptions*`*. These functions are documented on page* ??.)

`\@options` `\@options` is a synonym for `\ProcessOptions*` for upward compatibility with LaTeX2.09 style files.

```
\def\@options{\ProcessOptions*}
\@onlypreamble\@options
```

(*End definition for* `\@options`*. This function is documented on page* ??.)

`\@use@ption` Execute the code for the current option.

```
\def\@use@ption{%
  \@expandtwoargs\@removeelement\CurrentOption
  \@unusedoptionlist\@unusedoptionlist
  \csname ds@\CurrentOption\endcsname}
\@onlypreamble\@use@ption
```

(*End definition for* `\@use@ption`*. This function is documented on page* ??.)

`\ExecuteOptions` `\ExecuteOptions{`⟨*option-list*⟩`}` executes the code declared for each option.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease⟩                 {\ExecuteOptions}{Spaces in \ExecuteOptions}%
⟨*2ekernel | latexrelease⟩
\def\ExecuteOptions#1{%
```

Use \@fortmp here as it is anyway cleared during \@for loop so does not change any existing names.

```
  \edef\@fortmp{\zap@space#1 \@empty}%
  \def\reserved@a##1\@nil{%
    \@for\CurrentOption:=\@fortmp\do
             {\csname ds@\CurrentOption\endcsname}%
    \edef\CurrentOption{##1}}%
  \expandafter\reserved@a\CurrentOption\@nil}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\ExecuteOptions}{Spaces in \ExecuteOptions}%
⟨latexrelease⟩\def\ExecuteOptions#1{%
⟨latexrelease⟩  \def\reserved@a##1\@nil{%
⟨latexrelease⟩    \@for\CurrentOption:=#1\do
⟨latexrelease⟩             {\csname ds@\CurrentOption\endcsname}%
⟨latexrelease⟩    \edef\CurrentOption{##1}}%
⟨latexrelease⟩  \expandafter\reserved@a\CurrentOption\@nil}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@onlypreamble\ExecuteOptions
```

(*End definition for* \ExecuteOptions*. This function is documented on page* **??**.)

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
```

(*End definition for* \documentclass*. This function is documented on page* **??**.)

\documentstyle 2.09 style class 'style' declaration.

```
\def\documentstyle{%
  \makeatletter\input{latex209.def}\makeatother
  \documentclass}
\@onlypreamble\documentstyle
```

(*End definition for* \documentstyle*. This function is documented on page* **??**.)

\RequirePackage Load package if not already loaded.

```
\def\RequirePackage{%
  \@fileswithoptions\@pkgextension}
\@onlypreamble\RequirePackage
```

(*End definition for* \RequirePackage*. This function is documented on page* **??**.)

\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
```

(*End definition for* `\LoadClass`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@loadwithoptions`*. This function is documented on page* **??**.)

`\LoadClassWithOptions` Load class '`#1`' with the current option list.

```
\def\LoadClassWithOptions{%
  \@loadwithoptions\@clsextension\LoadClass}
\@onlypreamble\LoadClassWithOptions
```

(*End definition for* `\LoadClassWithOptions`*. This function is documented on page* **??**.)

`\RequirePackageWithOptions` Load package '`#1`' with the current option list.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\RequirePackageWithOptions}{Unused options issue}%
\def\RequirePackageWithOptions{%
```

The resetting of the unprocessed options is now done on a par package basis.

```
  \AtEndOfPackage{\expandafter\let
                     \csname unprocessedoptions-\@currname.\@currext\endcsname
                     \relax}%
  \@loadwithoptions\@pkgextension\RequirePackage}
\@onlypreamble\RequirePackageWithOptions
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\RequirePackageWithOptions}{Unused options issue}%
⟨latexrelease⟩
⟨latexrelease⟩\def\RequirePackageWithOptions{%
⟨latexrelease⟩  \AtEndOfPackage{\let\@unprocessedoptions\relax}%
⟨latexrelease⟩  \@loadwithoptions\@pkgextension\RequirePackage}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\RequirePackageWithOptions`*. This function is documented on page* **??**.)

\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
```

(*End definition for* \usepackage*. This function is documented on page* ??.)

\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
```

(*End definition for* \NeedsTeXFormat*. This function is documented on page* ??.)

\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}
```

(*End definition for* \zap@space*. This function is documented on page* ??.)

\@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.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease⟩         {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
⟨*2ekernel | latexrelease⟩
\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,{%
```

If #1 is \@nnil we have reached the end of the list (older version used \@nil here but \@nil is undefined so \ifx equal to all undefined commands)

```
      \ifx\@nnil##1\relax\else
```

If \ifx\@nnil##1\n@nil is true then #1 is (presumably) empty (Older code used \relax which is slighly easier to get into #1 by mistake, which would spoil this test.)

```
        \ifx\@nnil##1\@nnil\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,\@nnil,}%
  \fi
  \reserved@a}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩        {\@fileswith@pti@ns}{ifx tests in \@fileswith@pti@ns}%
⟨latexrelease⟩\def\@fileswith@pti@ns#1[#2]#3[#4]{%
⟨latexrelease⟩  \ifx#1\@clsextension
⟨latexrelease⟩    \ifx\@classoptionslist\relax
⟨latexrelease⟩      \xdef\@classoptionslist{\zap@space#2 \@empty}%
⟨latexrelease⟩      \def\reserved@a{%
⟨latexrelease⟩        \@onefilewithoptions#3[{#2}][{#4}]#1%
⟨latexrelease⟩        \@documentclasshook}%
⟨latexrelease⟩    \else
⟨latexrelease⟩      \def\reserved@a{%
⟨latexrelease⟩        \@onefilewithoptions#3[{#2}][{#4}]#1}%
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \else
⟨latexrelease⟩    \def\reserved@b##1,{%
⟨latexrelease⟩      \ifx\@nil##1\relax\else
⟨latexrelease⟩        \ifx\relax##1\relax\else
⟨latexrelease⟩         \noexpand\@onefilewithoptions##1[{#2}][{#4}]%
⟨latexrelease⟩         \noexpand\@pkgextension
⟨latexrelease⟩        \fi
⟨latexrelease⟩        \expandafter\reserved@b
⟨latexrelease⟩      \fi}%
⟨latexrelease⟩      \edef\reserved@a{\zap@space#3 \@empty}%
⟨latexrelease⟩      \edef\reserved@a{%
⟨latexrelease⟩        \expandafter\reserved@b\reserved@a,\@nil,}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \reserved@a}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
\@onlypreamble\@fileswith@pti@ns
```

This macro is used when loading packages or classes.

`\load@onefilewithoptions` Have the main argument as `#1`, so we only need one `\expandafter` above.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩      {\@onefilewithoptions}{Hooks and unused options issue}%
\def\@onefilewithoptions#1[#2][#3]#4{%
```

We have to sanitise file names, so that something like

```
\usepackage{some/local/path/array}
\usepackage{array}
```

won't load `array.sty` twice. It is remotely possible that those are two different files, but as a matter of principles, we will consider that the base file name uniquely identifies a

package, regardless of where it lives. This assumption already holds for file hooks, for example, which address the hook to a file by its base name only.

We'll use `\@expl@@@filehook@set@curr@file@@nNN` to parse the file name and return the ⟨*path*⟩ and ⟨*base+ext*⟩ in separate token lists. Further ahead, most operations use `\@currname` which doesn't have a path attached to it; only few actions prepend `\@currpath` to `\@currname` (namely loading, as we have to respect the given path).

A file substituiton isn't followed just yet because at this point we are parsing user input, so the file is still what the user asked for, and not the file actually loaded.

```
\@expl@@@filehook@set@curr@file@@nNN{#1.#4}\reserved@a\reserved@b
\edef\reserved@c{\def\noexpand\reserved@c####1%
  \detokenize\expandafter{\expanded{.#4}}%
  \noexpand\@nil{\def\noexpand\reserved@a{####1}}}\reserved@c
\expandafter\reserved@c\reserved@a\@nil
\@pushfilename
\xdef\@currname{\string@makeletter\reserved@a}%
\xdef\@currpath{\ifx\reserved@b\@empty\else\reserved@b/\fi}%
\global\let\@currext#4%
```

The command `\ver@`⟨*file*⟩`.`⟨*ext*⟩ is used to signal that a package is already loaded, either because it is in fact loaded, or because it's loading was suppressed. In minimal installations, said package may not exist but still have its loading suppressed with `\ver@`⟨*file*⟩`.`⟨*ext*⟩, so before checking if the file exists we have to check that we do need to load it with `\@ifl@aded`. If we don't, then there's no point in checking for a typo or load-disabling.

```
\@ifl@aded\@currext\@currname
```

If the package is already loaded, check that there were no option clashes:

```
  {\@if@ptions\@currext{\@currname}{#2}{}%
    {\@latex@error
      {Option clash for \@cls@pkg\space \@currname}%
      {The package \@currname\space has already been loaded
       with options:\MessageBreak
       \space\space[\@ptionlist{\@currname.\@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{\@currname.\@currext},#2\MessageBreak
       to your \noexpand\documentclass declaration may fix this.%
       \MessageBreak
       Try typing \space <return> \space to proceed.}}%
   \@firstofone}%
  {\makeatletter
```

The next line seems to be necessary for 2.09 compatibility (the way the code is written there) This seems questionable and should be look at as in 2e it is definitely unnecessary at this point!

```
    \@reset@ptions
```

First we take the ⟨*name*⟩ and ⟨*ext*⟩ given in the argument and check if the file exists, and issue an error otherwise asking for a correction with `\@missingfileerror`. For checking if the file exists we use `\@currpath` (usually empty) before `\@currname`.

```
    \IfFileExists{\@currpath\@currname.\@currext}{}%
```

```
        {\@missing@onefilewithoptions{#2}}%
```

If \@currname is empty (the user replied to the "Enter file name" prompt with ⟨*RETURN*⟩), so stop here (do \@popfilename to pop the item just added above).

This \@gobble omits the date check at the end.

```
      \ifx\@currname\@empty
        \expandafter\@gobble
      \else
```

If the file exists, check if it was load-prevented, and otherwise do the bookkeeping with \@filehook@file@push then call \set@curr@file to set \@curr@file (and do any required substitution), then actually load the class/package with \load@onefile@withoptions. \set@curr@file also needs the file path.

```
        \@disable@packageload@do{\@currname.\@currext}%
          {\@expl@@@filehook@file@push@@
           \set@curr@file{\@currpath\@currname.\@currext}%
           \@filehook@set@CurrentFile
```

The \set@curr@file line above might have replaced the file, so \@currname and \@currext may no longer hold the actual package being loaded, so in that case we need to update these two token lists (\@curr@file holds the file name after replacement, so we parse that).

The requested file is saved in \@currpkg@reqd to be used in \InputIfFileExists later: if the updated \@currname and \@currext are used we lose track of the substitution, so \CurrentFile and \CurrentFileUsed will be (incorrectly) the same.

```
           \expandafter\@swaptwoargs\expandafter
             {\expandafter{\@currpkg@reqd}}%
             {% <
```

\@currpkg@reqd doesn't take a path because it is used later to assign \opt@... and \ver@....

```
           \edef\@currpkg@reqd{\@currname.\@currext}%
           \ifx\CurrentFile\CurrentFileUsed
           \else
             \filename@parse\@curr@file
             \edef\@currpath{\string@makeletter\filename@area}%
             \edef\@currname{\string@makeletter\filename@base}%
             \edef\@currext{\string@makeletter\filename@ext}%
           \fi
           \load@onefile@withoptions{#2}%
           \def\@currpkg@reqd%{\@currpkg@reqd}
             }% >
```

Now just clean up and exit.

```
           \@expl@@@filehook@file@pop@@}%
        \expandafter\@firstofone
      \fi}%
```

Except in the case where \@currname is empty, the date is checked against the date marked in the package file:

```
    {\@ifl@ter\@currext{\@currname}{#3}{}%
      {\@latex@warning@no@line
        {You have requested,\on@line,
         version\MessageBreak
           `#3' of \@cls@pkg\space \@currname,\MessageBreak
```

```
        but only version\MessageBreak
         `\csname ver@\@currname.\@currext\endcsname'\MessageBreak
        is available}}%
    \ifx\@currext\@clsextension\let\LoadClass\@twoloadclasserror\fi}%
   \@popfilename
   \@reset@ptions}
\let\@currpkg@reqd\@empty
\@onlypreamble\@onefilewithoptions
```

The kernel no longer uses `\@unprocessedoptions`

```
\let\@unprocessedoptions\@undefined
```

`\@missing@onefilewithoptions`

Now the action taken when a file is not found. Path must be included here as it eventually leads to a file lookup.

```
\def\@missing@onefilewithoptions#1{%
  \@missingfileerror{\@currpath\@currname}\@currext
  \global\let\@currpath\@missingfile@area
  \global\let\@currname\@missingfile@base
  \global\let\@currext\@missingfile@ext}
```

`\load@onefile@withoptions`

Now the code that actually does the file loading:

```
\def\load@onefile@withoptions#1{%
  \let\CurrentOption\@empty
  \@reset@ptions
```

Grab everything in a macro, so the parameter stack is popped before any processing begins.

```
  \def\reserved@a{%
    \@pass@ptions\@currext{#1}{\@currname}%
    \expandafter\let
      \csname opt@\@currpkg@reqd\expandafter\endcsname
      \csname opt@\@currname.\@currext\endcsname
    \global\expandafter
    \let\csname ver@\@currname.\@currext\endcsname\@empty
```

We initialize `\...-h@@k` here and only if we load the file so that it remains undefined otherwise.

```
    \expandafter\let\csname\@currname.\@currext-h@@k\endcsname\@empty
```

When the current extension is `\@pkgextension` we are loading a package otherwise, if it is `\@clsextension`, a class, so depending on that we execute different hooks. If the extension is neither, then it is another type of file without special hooks.

```
%-----------------------------------------
    \ifx\@currext\@pkgextension
      \UseHook{package/before}%
      \UseHook{package/before/\@currname}%
    \else
      \ifx\@currext\@clsextension
        \UseHook{class/before}%
        \UseHook{class/before/\@currname}%
      \fi
    \fi
```

Now actually load the file (at this point we are certain it exists, but use `\InputIfFileExists` so that file hooks are executed). `\@currpath` is needed here too.

```
    \InputIfFileExists{\@currpath\@currpkg@reqd}{}%
      {\@latex@error
        {The \@cls@pkg\space\@currpkg@reqd\space failed to load}\@ehd}%
%----------------------------------------
```

In older versions of the code `\@unprocessedoptions` would generate an error for each specified option in a package unless a `\ProcessOptions` has appeared in the package file.

This has changed in 2020. We now use a separate macro per package to avoid interferences in case of nested packages. The whole code for handling this issue (GitHub 22) was provided by Hironobu Yamashita, thanks for that.

```
    \expandafter\let\csname unprocessedoptions-\@currname.\@currext\endcsname
                    \@@unprocessedoptions
    \csname\@currname.\@currext-h@@k\endcsname
    \expandafter\let\csname\@currname.\@currext-h@@k\endcsname
              \@undefined
```

Catch the case where the packages has handled the options and redefined `\@unprocessedoptions` to `\relax` (old interface). In that case no error should be produced.

```
    \ifx\@unprocessedoptions\relax
      \let\@unprocessedoptions\@undefined
```

Otherwise run the per package set of unused options.

```
    \else
      \csname unprocessedoptions-\@currname.\@currext\endcsname
    \fi
```

In either case we drop the macro afterwards as it is no longer needed.

```
    \expandafter\let
        \csname unprocessedoptions-\@currname.\@currext\endcsname
       \@undefined
```

And same procedure, James, when we are finished loading, except that the hook order is now reversed.

```
%----------------------------------------
    \ifx\@currext\@pkgextension
      \UseHook{package/after/\@currname}%
      \UseHook{package/after}%
    \else
      \ifx\@currext\@clsextension
        \UseHook{class/after/\@currname}%
        \UseHook{class/after}%
      \fi
    \fi}%
%----------------------------------------
  \@ifl@aded\@currext\@currname{}{\reserved@a}}
```

⟨/2ekernel | latexrelease⟩

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩      {\@onefilewithoptions}{Hooks and unused options issue}%
⟨latexrelease⟩
```

Because of the way `\@onfilewithoptions` is changed for rollback handling below we have to define `\load@onefilewithoptions` when rolling back!

```
⟨latexrelease⟩\def\load@onefilewithoptions#1[#2][#3]#4{%
⟨latexrelease⟩  \@pushfilename
⟨latexrelease⟩  \xdef\@currname{#1}%
⟨latexrelease⟩  \global\let\@currext#4%
⟨latexrelease⟩  \let\CurrentOption\@empty
⟨latexrelease⟩  \@reset@ptions
⟨latexrelease⟩  \makeatletter
⟨latexrelease⟩  \def\reserved@a{%
⟨latexrelease⟩    \@ifl@aded\@currext{#1}%
⟨latexrelease⟩      {\@if@ptions\@currext{#1}{#2}{}%
⟨latexrelease⟩        {\@latex@error
⟨latexrelease⟩            {Option clash for \@cls@pkg\space #1}%
⟨latexrelease⟩            {The package #1 has already been loaded
⟨latexrelease⟩             with options:\MessageBreak
⟨latexrelease⟩             \space\space[\@ptionlist{#1.\@currext}]\MessageBreak
⟨latexrelease⟩             There has now been an attempt to load it
⟨latexrelease⟩              with options\MessageBreak
⟨latexrelease⟩             \space\space[#2]\MessageBreak
⟨latexrelease⟩             Adding the global options:\MessageBreak
⟨latexrelease⟩             \space\space
⟨latexrelease⟩                  \@ptionlist{#1.\@currext},#2\MessageBreak
⟨latexrelease⟩             to your \noexpand\documentclass declaration may fix this.%
⟨latexrelease⟩             \MessageBreak
⟨latexrelease⟩             Try typing \space <return> \space to proceed.}}}%
⟨latexrelease⟩      {\@pass@ptions\@currext{#2}{#1}%
⟨latexrelease⟩       \global\expandafter
⟨latexrelease⟩       \let\csname ver@\@currname.\@currext\endcsname\@empty
⟨latexrelease⟩       \expandafter\let\csname\@currname.\@currext-h@@k\endcsname\@empty
⟨latexrelease⟩       \InputIfFileExists
⟨latexrelease⟩         {\@currname.\@currext}%
⟨latexrelease⟩         {}%
⟨latexrelease⟩         {\@missingfileerror\@currname\@currext}%
⟨latexrelease⟩    \let\@unprocessedoptions\@@unprocessedoptions
⟨latexrelease⟩    \csname\@currname.\@currext-h@@k\endcsname
⟨latexrelease⟩    \expandafter\let\csname\@currname.\@currext-h@@k\endcsname
⟨latexrelease⟩              \@undefined
⟨latexrelease⟩    \@unprocessedoptions}%
⟨latexrelease⟩    \@ifl@ter\@currext{#1}{#3}{}%
⟨latexrelease⟩      {\@latex@warning@no@line
⟨latexrelease⟩         {You have requested,\on@line,
⟨latexrelease⟩          version\MessageBreak
⟨latexrelease⟩            `#3' of \@cls@pkg\space #1,\MessageBreak
⟨latexrelease⟩          but only version\MessageBreak
⟨latexrelease⟩           `\csname ver@#1.\@currext\endcsname'\MessageBreak
⟨latexrelease⟩          is available}}%
⟨latexrelease⟩    \ifx\@currext\@clsextension\let\LoadClass\@twoloadclasserror\fi
⟨latexrelease⟩    \@popfilename
⟨latexrelease⟩    \@reset@ptions}%
⟨latexrelease⟩  \reserved@a}
⟨latexrelease⟩
⟨latexrelease⟩\let \load@onefile@withoptions    \@undefined
⟨latexrelease⟩\let \@missing@onefilewithoptions \@undefined
```

```
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@fileswithoptions` *and others. These functions are documented on page* **??**.)

`\@@fileswith@pti@ns` Save the definition (for error checking).

```
\let\@@fileswith@pti@ns\@fileswith@pti@ns
\@onlypreamble\@@fileswith@pti@ns
```

(*End definition for* `\@@fileswith@pti@ns`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@reset@ptions`*. This function is documented on page* **??**.)

4.1 Hooks

Allow code to 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 beginning or end of the document.

```
\ifx\@begindocumenthook\@undefined
  \let\@begindocumenthook\@empty
\fi
\let\@enddocumenthook\@empty
```

(*End definition for* `\@begindocumenthook` *and* `\@enddocumenthook`*. These functions are documented on page* **??**.)

`\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
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\AtBeginDocument}{Use hook system}%
\DeclareRobustCommand\AtBeginDocument{\AddToHook{begindocument}}
\DeclareRobustCommand\AtEndDocument  {\AddToHook{enddocument}}
%\DeclareRobustCommand\AtEndDocument {\AddToHook{env/document/end}} % alternative impl
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\AtBeginDocument}{Use hook system}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand\AtBeginDocument{\g@addto@macro\@begindocumenthook}
⟨latexrelease⟩\DeclareRobustCommand\AtEndDocument{\g@addto@macro\@enddocumenthook}
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\@onlypreamble\AtBeginDocument
```

(*End definition for* `\AtEndOfPackage` *and others. These functions are documented on page* **??**.)

`\@cls@pkg` The current file type.

```
\def\@cls@pkg{%
  \ifx\@currext\@clsextension
    document class%
  \else
    package%
  \fi}
\@onlypreamble\@cls@pkg
```

(*End definition for* `\@cls@pkg`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@unknownoptionerror`*. This function is documented on page* **??**.)

`\@@unprocessedoptions` Declare an error for each option, unless a `\ProcessOptions` occurred.

```
\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
```

(*End definition for* `\@@unprocessedoptions`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@badrequireerror`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@twoloadclasserror`*. This function is documented on page* **??**.)

`\@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
```

(*End definition for* `\@twoclasseserror`*. This function is documented on page* **??**.)

4.2 Providing shipment

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

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

(*End definition for* `\two@digits`*. This function is documented on page* **??**.)

`\filecontents` `\endfilecontents` This environment implements inline files. The star-form does not write extra comments into the file.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\filec@ntents}{Define \q@curr@file directly (gh/220)}%
%
```

We use `@tempswa` to mean no preamble writing and reuse `@filesw` to indicate no overwriting:

```
\def\filecontents{\@tempswatrue\@fileswtrue
  \@ifnextchar[\filec@ntents@opt\filec@ntents
}
\@namedef{filecontents*}{\@tempswafalse\@fileswtrue
  \@ifnextchar[\filec@ntents@opt\filec@ntents
}
```

To handle the optional argument we execute for each option the command `\filec@ntents@OPTION` if it exist or complain about unknown option.

```
\def\filec@ntents@opt[#1]{%
  \edef\@fortmp{\zap@space#1 \@empty}%
  \@for\reserved@a:=\@fortmp\do{%
    \ifcsname filec@ntents@\reserved@a\endcsname
```

```
      \csname filec@ntents@\reserved@a\endcsname
    \else
    \@latex@error{Unknown filecontents option \reserved@a}%
       {Valid options are force (or overwrite), nosearch, noheader}%
    \fi}%
  \filec@ntents
}
```

Option `force` (or `overwrite`) changes the overwriting switch

```
\let\filec@ntents@force\@fileswfalse
\let\filec@ntents@overwrite\@fileswfalse  % alternative name
```

and option `noheader` the preamble switch (which is equivalent to using the star form of the environment).

```
\let\filec@ntents@noheader\@tempswafalse
```

Option `nosearch` only checks the current directory not the whole TEX tree for the existence of the file to write.

```
\def\filec@ntents@nosearch{%
  \let\filec@ntents@checkdir\@currdir
  \def\filec@ntents@where{in current directory}}
```

By default we search the whole tree:

```
\let\filec@ntents@checkdir\@empty
\def\filec@ntents@where{exists on the system}
```

```
\begingroup%
\@tempcnta=1
\loop
  \catcode\@tempcnta=12  %
  \advance\@tempcnta\@ne %
\ifnum\@tempcnta<32      %
\repeat                  %
\catcode`\*=11 %
\catcode`\^^M\active%
\catcode`\^^L\active\let^^L\relax%
\catcode`\^^I\active%
```

```
\gdef\filec@ntents#1{%
  \set@curr@file{\filec@ntents@checkdir#1}%
  \edef\q@curr@file{"\@curr@file"}%
```

LuaTEX has more writes (and 18 is safe here).

```
  \chardef\reserved@c\ifx\directlua\@undefined 15 \else 127 \fi%
  \openin\@inputcheck\q@curr@file \space %
  \ifeof\@inputcheck%
    \@latex@warning@no@line%
        {Writing file `\@currdir\@curr@file'}%
```

```
    \ch@ck7\reserved@c\write\relax%
    \immediate\openout\reserved@c\q@curr@file\relax%
  \else%
```

```
    \if@filesw%
      \@latex@warning@no@line%
          {File `\@curr@file' already \filec@ntents@where.\MessageBreak%
             Not generating it from this source}%
```

```
      \let\write\@gobbletwo%
      \let\closeout\@gobble%
    \else%
```

If we are overwriting, we try to make sure that the user is not by mistake overwriting the input file (`\jobname`). Of course, this only works for input files ending in `.tex`. If a different extension is used there is no way to see that we are overwriting ourselves!

```
      \edef\reserved@a{#1}%
      \edef\reserved@a{\detokenize\expandafter{\reserved@a}}%
      \edef\reserved@b{\detokenize\expandafter{\jobname}}%
      \ifx\reserved@a\reserved@b%
        \@fileswtrue%
      \else%
        \edef\reserved@b{\reserved@b\detokenize{.tex}}%
        \ifx\reserved@a\reserved@b
          \@fileswtrue%
        \fi%
      \fi%
```

We allocate a write channel but we open it only if it is (hopefully) safe. If not opened that means we are going to write on the terminal.

```
      \ch@ck7\reserved@c\write\relax%
      \if@filesw%  % Foul ... trying to overwrite \jobname!
      \@latex@error{Trying to overwrite `\jobname.tex'}{You can't %
        write to the file you are reading from!\MessageBreak%
        Data is written to screen instead.}%
      \else%
        \@latex@warning@no@line%
           {Writing or overwriting file `\@currdir\@curr@file'}%
        \immediate\openout\reserved@c\q@curr@file\relax%
      \fi%
    \fi%
  \fi%
```

Closing the `\@inputcheck` is done here to avoid having to do this in each branch.

```
  \closein\@inputcheck%
  \if@tempswa%

    \immediate\write\reserved@c{%
      \@percentchar\@percentchar\space%
          \expandafter\@gobble\string\LaTeX2e file `\@curr@file'^^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%
```

If there are active characters in the upper half (e.g., from `inputenc` there would be confusion so we render everything harmless.

```
  \count@ 128\relax%
  \loop%
    \catcode\count@ 11\relax%
    \advance\count@ \@ne%
```

```
    \ifnum\count@<\@cclvi%
  \repeat%
```

```
  \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 \@curr@file}%
        \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{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
  \catcode‘\^^I\active%
  \let\I\@undefined%
  \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
  \catcode‘\^^M\active%
  \edef^^M##1^^M{%
    \noexpand\reserved@b##1\E\E\relax}}%
\endgroup%
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩                  {\filec@ntents}{Spaces in file names + optional arg}%
⟨latexrelease⟩\def\filecontents{\@tempswatrue\@fileswtrue
⟨latexrelease⟩  \@ifnextchar[\filec@ntents@opt\filec@ntents
⟨latexrelease⟩}
⟨latexrelease⟩\@namedef{filecontents*}{\@tempswafalse\@fileswtrue
⟨latexrelease⟩  \@ifnextchar[\filec@ntents@opt\filec@ntents
⟨latexrelease⟩}
⟨latexrelease⟩\def\filec@ntents@opt[#1]{%
⟨latexrelease⟩  \edef\@fortmp{\zap@space#1 \@empty}%
```

```
⟨latexrelease⟩  \@for\reserved@a:=\@fortmp\do{%
⟨latexrelease⟩    \ifcsname filec@ntents@\reserved@a\endcsname
⟨latexrelease⟩      \csname filec@ntents@\reserved@a\endcsname
⟨latexrelease⟩    \else
⟨latexrelease⟩    \@latex@error{Unknown filecontents option \reserved@a}%
⟨latexrelease⟩       {Valid options are force (or overwrite), nosearch, noheader}%
⟨latexrelease⟩    \fi}%
⟨latexrelease⟩  \filec@ntents
⟨latexrelease⟩}
⟨latexrelease⟩\let\filec@ntents@force\@fileswfalse
⟨latexrelease⟩\let\filec@ntents@overwrite\@fileswfalse  % alternative name
⟨latexrelease⟩\let\filec@ntents@noheader\@tempswafalse
⟨latexrelease⟩\def\filec@ntents@nosearch{%
⟨latexrelease⟩  \let\filec@ntents@checkdir\@currdir
⟨latexrelease⟩  \def\filec@ntents@where{in current directory}}
⟨latexrelease⟩\let\filec@ntents@checkdir\@empty
⟨latexrelease⟩\def\filec@ntents@where{exists on the system}
⟨latexrelease⟩\begingroup%
⟨latexrelease⟩\@tempcnta=1
⟨latexrelease⟩\loop
⟨latexrelease⟩  \catcode\@tempcnta=12  %
⟨latexrelease⟩  \advance\@tempcnta\@ne %
⟨latexrelease⟩\ifnum\@tempcnta<32      %
⟨latexrelease⟩\repeat                  %
⟨latexrelease⟩\catcode`\*=11 %
⟨latexrelease⟩\catcode`\^^M\active%
⟨latexrelease⟩\catcode`\^^L\active\let^^L\relax%
⟨latexrelease⟩\catcode`\^^I\active%
⟨latexrelease⟩\gdef\filec@ntents#1{%
⟨latexrelease⟩  \set@curr@file{\filec@ntents@checkdir#1}%
⟨latexrelease⟩  \edef\q@curr@file{\expandafter\quote@name\expandafter{\@curr@file}}%
⟨latexrelease⟩  \chardef\reserved@c\ifx\directlua\@undefined 15 \else 127 \fi%
⟨latexrelease⟩  \openin\@inputcheck\q@curr@file \space %
⟨latexrelease⟩  \ifeof\@inputcheck%
⟨latexrelease⟩    \@latex@warning@no@line%
⟨latexrelease⟩        {Writing file `\@currdir\@curr@file'}%
⟨latexrelease⟩    \ch@ck7\reserved@c\write\relax%
⟨latexrelease⟩    \immediate\openout\reserved@c\q@curr@file\relax%
⟨latexrelease⟩  \else%
⟨latexrelease⟩    \if@filesw%
⟨latexrelease⟩      \@latex@warning@no@line%
⟨latexrelease⟩          {File `\@curr@file' already \filec@ntents@where.\MessageBreak%
⟨latexrelease⟩             Not generating it from this source}%
⟨latexrelease⟩      \let\write\@gobbletwo%
⟨latexrelease⟩      \let\closeout\@gobble%
⟨latexrelease⟩    \else%
⟨latexrelease⟩      \edef\reserved@a{#1}%
⟨latexrelease⟩      \edef\reserved@a{\detokenize\expandafter{\reserved@a}}%
⟨latexrelease⟩      \edef\reserved@b{\detokenize\expandafter{\jobname}}%
⟨latexrelease⟩      \ifx\reserved@a\reserved@b%
⟨latexrelease⟩        \@fileswtrue%
⟨latexrelease⟩      \else%
⟨latexrelease⟩        \edef\reserved@b{\reserved@b\detokenize{.tex}}%
⟨latexrelease⟩        \ifx\reserved@a\reserved@b
```

```
⟨latexrelease⟩          \@fileswtrue%
⟨latexrelease⟩        \fi%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩      \ch@ck7\reserved@c\write\relax%
⟨latexrelease⟩      \if@filesw%  % Foul ... trying to overwrite \jobname!
⟨latexrelease⟩      \@latex@error{Trying to overwrite ‘\jobname.tex’}{You can’t %
⟨latexrelease⟩        write to the file you are reading from!\MessageBreak%
⟨latexrelease⟩        Data is written to screen instead.}%
⟨latexrelease⟩      \else%
⟨latexrelease⟩        \@latex@warning@no@line%
⟨latexrelease⟩           {Writing or overwriting file ‘\@currdir\@curr@file’}%
⟨latexrelease⟩        \immediate\openout\reserved@c\q@curr@file\relax%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩    \fi%
⟨latexrelease⟩  \fi%
⟨latexrelease⟩  \closein\@inputcheck%
⟨latexrelease⟩  \if@tempswa%
⟨latexrelease⟩    \immediate\write\reserved@c{%
⟨latexrelease⟩      \@percentchar\@percentchar\space%
⟨latexrelease⟩          \expandafter\@gobble\string\LaTeX2e file ‘\@curr@file’^^J%
⟨latexrelease⟩      \@percentchar\@percentchar\space  generated by the %
⟨latexrelease⟩        ‘\@currenvir’ \expandafter\@gobblefour\string\newenvironment^^J%
⟨latexrelease⟩      \@percentchar\@percentchar\space from source ‘\jobname’ on %
⟨latexrelease⟩         \number\year/\two@digits\month/\two@digits\day.^^J%
⟨latexrelease⟩      \@percentchar\@percentchar}%
⟨latexrelease⟩  \fi%
⟨latexrelease⟩  \let\do\@makeother\dospecials%
⟨latexrelease⟩  \count@ 128\relax%
⟨latexrelease⟩  \loop%
⟨latexrelease⟩    \catcode\count@ 11\relax%
⟨latexrelease⟩    \advance\count@ \@ne%
⟨latexrelease⟩    \ifnum\count@<\@cclvi%
⟨latexrelease⟩  \repeat%
⟨latexrelease⟩  \edef\E{\@backslashchar end\string{\@currenvir\string}}%
⟨latexrelease⟩  \edef\reserved@b{%
⟨latexrelease⟩    \def\noexpand\reserved@b%
⟨latexrelease⟩         ####1\E####2\E####3\relax}%
⟨latexrelease⟩  \reserved@b{%
⟨latexrelease⟩    \ifx\relax##3\relax%
⟨latexrelease⟩      \immediate\write\reserved@c{##1}%
⟨latexrelease⟩    \else%
⟨latexrelease⟩      \edef^^M{\noexpand\end{\@currenvir}}%
⟨latexrelease⟩      \ifx\relax##1\relax%
⟨latexrelease⟩      \else%
⟨latexrelease⟩          \@latex@warning{Writing text ‘##1’ before %
⟨latexrelease⟩             \string\end{\@currenvir}\MessageBreak as last line of \@curr@file}%
⟨latexrelease⟩        \immediate\write\reserved@c{##1}%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩      \ifx\relax##2\relax%
⟨latexrelease⟩      \else%
⟨latexrelease⟩         \@latex@warning{%
⟨latexrelease⟩           Ignoring text ‘##2’ after \string\end{\@currenvir}}%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩    \fi%
```

```
⟨latexrelease⟩    ^^M}%
⟨latexrelease⟩  \catcode`\^^L\active%
⟨latexrelease⟩  \let\L\@undefined%
⟨latexrelease⟩  \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
⟨latexrelease⟩  \catcode`\^^I\active%
⟨latexrelease⟩  \let\I\@undefined%
⟨latexrelease⟩  \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
⟨latexrelease⟩  \catcode`\^^M\active%
⟨latexrelease⟩  \edef^^M##1^^M{%
⟨latexrelease⟩    \noexpand\reserved@b##1\E\E\relax}}%
⟨latexrelease⟩\endgroup%
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\filec@ntents}{Spaces in file names + optional arg}%
⟨latexrelease⟩
⟨latexrelease⟩\let\filec@ntents@opt        \@undefined
⟨latexrelease⟩\let\filec@ntents@force      \@undefined
⟨latexrelease⟩\let\filec@ntents@overwrite  \@undefined
⟨latexrelease⟩\let\filec@ntents@noheader   \@undefined
⟨latexrelease⟩\let\filec@ntents@nosearch   \@undefined
⟨latexrelease⟩\let\filec@ntents@checkdir   \@undefined
⟨latexrelease⟩\let\filec@ntents@where      \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\begingroup%
⟨latexrelease⟩\@tempcnta=1
⟨latexrelease⟩\loop
⟨latexrelease⟩  \catcode\@tempcnta=12  %
⟨latexrelease⟩  \advance\@tempcnta\@ne %
⟨latexrelease⟩\ifnum\@tempcnta<32      %
⟨latexrelease⟩\repeat                  %
⟨latexrelease⟩\catcode`\*=11 %
⟨latexrelease⟩\catcode`\^^M\active%
⟨latexrelease⟩\catcode`\^^L\active\let^^L\relax%
⟨latexrelease⟩\catcode`\^^I\active%
⟨latexrelease⟩
⟨latexrelease⟩\gdef\filec@ntents#1{%
⟨latexrelease⟩  \openin\@inputcheck#1 %
⟨latexrelease⟩  \ifeof\@inputcheck%
⟨latexrelease⟩    \@latex@warning@no@line%
⟨latexrelease⟩        {Writing file `\@currdir#1'}%
⟨latexrelease⟩    \chardef\reserved@c15 %
⟨latexrelease⟩    \ch@ck7\reserved@c\write%
⟨latexrelease⟩    \immediate\openout\reserved@c#1\relax%
⟨latexrelease⟩  \else%
⟨latexrelease⟩    \closein\@inputcheck%
⟨latexrelease⟩    \@latex@warning@no@line%
⟨latexrelease⟩            {File `#1' already exists on the system.\MessageBreak%
⟨latexrelease⟩             Not generating it from this source}%
⟨latexrelease⟩    \let\write\@gobbletwo%
⟨latexrelease⟩    \let\closeout\@gobble%
⟨latexrelease⟩  \fi%
⟨latexrelease⟩  \if@tempswa%
⟨latexrelease⟩    \immediate\write\reserved@c{%
⟨latexrelease⟩      \@percentchar\@percentchar\space%
```

```
⟨latexrelease⟩          \expandafter\@gobble\string\LaTeX2e file ‘#1’^^J%
⟨latexrelease⟩      \@percentchar\@percentchar\space  generated by the %
⟨latexrelease⟩        ‘\@currenvir’ \expandafter\@gobblefour\string\newenvironment^^J%
⟨latexrelease⟩      \@percentchar\@percentchar\space from source ‘\jobname’ on %
⟨latexrelease⟩         \number\year/\two@digits\month/\two@digits\day.^^J%
⟨latexrelease⟩      \@percentchar\@percentchar}%
⟨latexrelease⟩  \fi%
⟨latexrelease⟩  \let\do\@makeother\dospecials%
⟨latexrelease⟩  \count@ 128\relax%
⟨latexrelease⟩  \loop%
⟨latexrelease⟩    \catcode\count@ 11\relax%
⟨latexrelease⟩    \advance\count@ \@ne%
⟨latexrelease⟩    \ifnum\count@<\@cclvi%
⟨latexrelease⟩  \repeat%
⟨latexrelease⟩  \edef\E{\@backslashchar end\string{\@currenvir\string}}%
⟨latexrelease⟩  \edef\reserved@b{%
⟨latexrelease⟩    \def\noexpand\reserved@b%
⟨latexrelease⟩         ####1\E####2\E####3\relax}%
⟨latexrelease⟩  \reserved@b{%
⟨latexrelease⟩    \ifx\relax##3\relax%
⟨latexrelease⟩      \immediate\write\reserved@c{##1}%
⟨latexrelease⟩    \else%
⟨latexrelease⟩      \edef^^M{\noexpand\end{\@currenvir}}%
⟨latexrelease⟩      \ifx\relax##1\relax%
⟨latexrelease⟩      \else%
⟨latexrelease⟩          \@latex@warning{Writing text ‘##1’ before %
⟨latexrelease⟩             \string\end{\@currenvir}\MessageBreak as last line of #1}%
⟨latexrelease⟩        \immediate\write\reserved@c{##1}%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩      \ifx\relax##2\relax%
⟨latexrelease⟩      \else%
⟨latexrelease⟩         \@latex@warning{%
⟨latexrelease⟩           Ignoring text ‘##2’ after \string\end{\@currenvir}}%
⟨latexrelease⟩      \fi%
⟨latexrelease⟩    \fi%
⟨latexrelease⟩    ^^M}%
⟨latexrelease⟩
⟨latexrelease⟩  \catcode‘\^^L\active%
⟨latexrelease⟩  \let\L\@undefined%
⟨latexrelease⟩  \def^^L{\expandafter\ifx\csname L\endcsname\relax\fi ^^J^^J}%
⟨latexrelease⟩  \catcode‘\^^I\active%
⟨latexrelease⟩  \let\I\@undefined%
⟨latexrelease⟩  \def^^I{\expandafter\ifx\csname I\endcsname\relax\fi\space}%
⟨latexrelease⟩  \catcode‘\^^M\active%
⟨latexrelease⟩  \edef^^M##1^^M{%
⟨latexrelease⟩    \noexpand\reserved@b##1\E\E\relax}}%
⟨latexrelease⟩\endgroup%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\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
```

We no longer prevent the code to be used after begin document (no rollback needed for this change).

```
%\@onlypreamble\filecontents
%\@onlypreamble\endfilecontents
%\@onlypreamble\filecontents*
%\@onlypreamble\endfilecontents*
\endgroup
%\@onlypreamble\filec@ntents
```

(*End definition for* `\filecontents` *and* `\endfilecontents`*. These functions are documented on page* **??**.)

5 Package/class rollback mechanism

```
⟨/2ekernel⟩
⟨*2ekernel | latexreleasefirst⟩
```

`\pkgcls@debug` For testing we have a few extra lines of code that by default do nothing but one can set `\pkgcls@debug` to `\typeout` to get extra info. Sometime in the future this will be dropped.

```
⟨*tracerollback⟩
%\let\pkgcls@debug\typeout
\let\pkgcls@debug\@gobble
⟨/tracerollback⟩
```

(*End definition for* `\pkgcls@debug`*. This function is documented on page* **??**.)

`\requestedLaTeXdate` The macro (!) `\requestedLaTeXdate` holds the globally requested rollback date (via `latexrelease`) or zero if no such request was made.

```
\def\requestedLaTeXdate{0}
```

(*End definition for* `\requestedLaTeXdate`*. This function is documented on page* **??**.)

`\pkgcls@targetdate` `\pkgcls@targetlabel` `\pkgcls@innerdate` If a rollback for a package or class is requested then `\pkgcls@targetdate` holds the requested date as a number YYYYMMDD (if there was one, otherwise the value of `\requestedLaTeXdate`) and `\pkgcls@targetlabel` will be empty. If there was a request for a named version then `\pkgcls@targetlabel` holds the verion name and `\pkgcls@targetdate` is set to `1`.

`\pkgcls@targetdate=0` is used to indicate that there was no rollback request. While loading an old release `\pkgcls@targetdate` is also reset to zero so that `\DeclareRelease` declarations are bypassed.

In contrast `\pkgcls@innerdate` will always hold the requested date (in a macro not a counter) if there was one, otherwise, e.g., if there was no request or a request to a

version name it will contain TEX largest legal number. While loading a file this can be used to provide conditionals that select code based on the request.

```
\ifx\pkgcls@targetdate\@undefined
  \newcount\pkgcls@targetdate
\fi
\let\pkgcls@targetlabel\@empty
\def\pkgcls@innerdate{\maxdimen}
```

(*End definition for* `\pkgcls@targetdate`, `\pkgcls@targetlabel`, *and* `\pkgcls@innerdate`. *These functions are documented on page* ??.)

`\pkgcls@candidate`
`\pkgcls@releasedate`

When looping through the `\DeclareRelease` declarations we record if the release is the best candidate we have seen so far. This is recorded in `\pkgcls@candidate` and we update it whenever we see a better one.

In `\pkgcls@releasedate` we keep track of the release date of that candidate.

```
\let\pkgcls@candidate\@empty
\let\pkgcls@releasedate\@empty
```

(*End definition for* `\pkgcls@candidate` *and* `\pkgcls@releasedate`. *These functions are documented on page* ??.)

`\load@onefilewithoptions`
`\@onefilewithoptions`

the best place to add the rollback code is at the point where `\@onefilewithoptions` is called to load a single class or package.

To make things easy we save the old definition as `\load@onefilewithoptions` and then provide a new interface.

Important: as this code is also unconditionally placed into latexrelease we can only do this name change once otherwise both macros will contain the same code.

```
\ifx\load@onefilewithoptions\@undefined
 \let\load@onefilewithoptions\@onefilewithoptions
```

```
 \def\@onefilewithoptions#1[#2][#3]#4{%
```

First a bit of tracing normally disabled.

```
⟨*tracerollback⟩
  \pkgcls@debug{--- File loaded request (\noexpand\usepackage or ...)}%
  \pkgcls@debug{\@spaces 1: #1}%
  \pkgcls@debug{\@spaces 2: #2}%
  \pkgcls@debug{\@spaces 3: #3}%
  \pkgcls@debug{\@spaces 4: #4}%
⟨/tracerollback⟩
```

Two of the arguments are needed later on in error/warning messages so we save them.

```
  \def\pkgcls@name{#1}%                  % for info message
  \def\pkgcls@arg {#3}%                  % for info message
```

then we parse the final optional argument to determine if there is a specific rollback request for the current file. This will set `\pkgcls@targetdate`, `\pkgcls@targetlabel` and `\pkgcls@mindate`.

```
  \pkgcls@parse@date@arg{#3}%
```

When determining the correct release to load we keep track of candiates in `\pkgcls@candidate` and initially we don't have any:

```
  \let\pkgcls@candidate\@empty
```

If we had a rollback request then #3 may contain data but not necessarily a "minimal date" so instead of passing it on we pass on the content of \pkgcls@mindate. We need to pass the value not the command, otherwise nested packages may pick up the wrong information.

```
  \begingroup
  \edef\reserved@a{%
    \endgroup
    \unexpanded{\load@onefilewithoptions#1[#2]}%
    [\pkgcls@mindate]%
    \unexpanded{#4}}%
   \reserved@a
 }
\fi
```

(*End definition for* \load@onefilewithoptions *and* \@onefilewithoptions*. These functions are documented on page* ??.)

\pkgcls@parse@date@arg The \pkgcls@parse@date@arg command parses the second optional argument of \usepackage, \RequirePackage or \documentclass for a rollback request setting the values of \pkgcls@targetdate and \pkgcls@targetlabel.

This optional argument has a dual purpose: If it just contains a date string then this means that the package should have at least that date (to ensure that a certain feature is actually available, or a certain bug has been fixed). When the package gets loaded the information in \Provides... will then be checked against this request.

But if it starts with an equal sign followed by a date string or followed by a version name then this means that we should roll back to the state of the package at tht date or to the version with the requested name.

If there was no optional argument or the optional argument does not start with "=" then the \pkgcls@targetdate is set to the date of the overall rollback request (via latexrelease) or if that was not given it is set to 0. In either case \pkgcls@targetlabel will be made empty.

If the argument doesn't start with "=" then it is supposed to be a "minimal date" and we therefore save the value in \pkgcls@mindate, otherwise this macro is made empty.

So in summary we have:

Input		\pkgcls@targetdate	\pkgcls@targetlabel	\pkgcls@mindate
⟨*empty*⟩	→	⟨*global-rollbackdate-as-number*⟩	⟨*empty*⟩	⟨*empty*⟩
⟨*date*⟩	→	⟨*global-rollbackdate-as-number*⟩	⟨*empty*⟩	⟨*date*⟩
=⟨*date*⟩	→	⟨*date-as-number*⟩	⟨*empty*⟩	⟨*empty*⟩
=⟨*version*⟩	→	1	⟨*version*⟩	⟨*empty*⟩
⟨*other*⟩	→	⟨*global-rollbackdate-as-number*⟩	⟨*empty*⟩	⟨*other*⟩

where ⟨*global-rollbackdate-as-number*⟩ is a date request given via latexrelease or if there wasn't one 0.

```
\def\pkgcls@parse@date@arg #1{%
```

If the argument is empty we use the rollback date from latexrelease which has the value of zero if there was no rollback request. The label and the minimal date is made empty in that case.

```
  \ifx\@nil#1\@nil
    \pkgcls@targetdate\requestedLaTeXdate\relax
    \let\pkgcls@targetlabel\@empty
    \let\pkgcls@mindate\@empty
```

Otherwise we parse the argument further, checking for a = as the first character. We append a = at the end so that there is at least one such character in the argument.

```
  \else
    \pkgcls@parse@date@arg@#1=\@nil\relax
  \fi
}
```

The actual parsing work then happens in \pkgcls@parse@date@arg@:

```
\def\pkgcls@parse@date@arg@#1=#2\@nil{%
```

We set \pkgcls@targetdate depending on the parsing result; the code is expandable so we can do the parsing as part of the assignment.

```
 \pkgcls@targetdate
```

If a = was in first position then #1 will be empty. In that case #2 will be the original argument with a = appended.

This can be parsed with \@parse@version, the trailing character is simply ignored. This macro returns the parsed date as a number (or zero if it wasn't a date) and accepts both YYYY/MM/DD and YYYY-MM-DD formats.

```
   \ifx\@nil#1\@nil
     \@parse@version0#2//00\@nil\relax
```

Whatever is returned is thus assigned to \pkgcls@targetdate and therefore we can now test its value. If the value is zero we assume that the remaining argument string represents a version and change \pkgcls@targetdate and set \pkgcls@targetlabel to the version name (after stripping off the trailing =.

```
     \ifnum \pkgcls@targetdate=\z@
       \pkgcls@targetdate\@ne
       \def\pkgcls@innerdate{\maxdimen}%
       \pkgcls@parse@date@arg@version#2%
     \else
       \edef\pkgcls@innerdate{\the\pkgcls@targetdate}%
     \fi
     \let\pkgcls@mindate\@empty
   \else
```

If #1 was not empty then there wasn't a = character in first position so we are dealing either with a "minimum date" or with some incorrect data. We assume the former and make the following assignments (the first one finishing the assignment of \pkgcls@targetdate):

```
     \requestedLaTeXdate\relax
     \let\pkgcls@targetlabel\@empty
     \def\pkgcls@innerdate{\maxdimen}%
     \def\pkgcls@mindate{#1}%
```

If the min-date is after the requested rollback date (if there is any, i.e., if it is not zero) then we have a conflict and therefore issue a warning.

```
     \ifnum \pkgcls@targetdate > \z@
       \ifnum \@parse@version0#1//00\@nil > \pkgcls@targetdate
         \@latex@warning@no@line{Suspicious rollback/min-date date given\MessageBreak
           A minimal date of #1 has been specified for
            \@cls@pkg\MessageBreak '\pkgcls@name'.\MessageBreak
            But this is in conflict
            with a rollback request to \requestedpatchdate}
       \fi
```

```
      \fi
    \fi
}
```

Strip off the trailing = and assign the version name to \pkgcls@targetlabel.

```
\def\pkgcls@parse@date@arg@version#1={%
  \def\pkgcls@targetlabel{#1}}
```

(*End definition for* \pkgcls@parse@date@arg. *This function is documented on page* ??.)

\DeclareRelease First argument is the "name" of the release and it can be left empty if one doesn't like to give a name to the release. The second argument is that from which on this release was available (or should be used in case of minor updates). The final argument is the external file name of this release, by convention this should be ⟨*pkg/cls-name*⟩-⟨*date*⟩.⟨*extension*⟩ but this is not enforced and through this argument one can overwrite it.

```
\def\DeclareRelease#1#2#3{%
  \ifnum\pkgcls@targetdate>\z@  % some sort of rollback request
⟨*tracerollback⟩
    \pkgcls@debug{---\string\DeclareRelease:}%
    \pkgcls@debug{\@spaces 1: #1}%
    \pkgcls@debug{\@spaces 2: #2}%
    \pkgcls@debug{\@spaces 3: #3}%
⟨/tracerollback⟩
```

If the date argument #2 is empty we are dealing with a special release that should be only accessible via its name; a typical use case would be a "beta" release. So if we are currently processing a date request we ignore it and otherwise we check if we can match the name and if so load the corresponding release file.

```
    \ifx\@nil#2\@nil
      \ifnum\pkgcls@targetdate=\@ne  % named request
        \def\reserved@a{#1}%
        \ifx\pkgcls@targetlabel\reserved@a
          \pkgcls@use@this@release{#3}{}%
⟨*tracerollback⟩
        \else
          \pkgcls@debug{Label doesn't match}%
⟨/tracerollback⟩
        \fi
⟨*tracerollback⟩
      \else
        \pkgcls@debug{Date request: ignored}%
⟨/tracerollback⟩
      \fi
    \else
```

If the value of \pkgcls@targetdate is greater than 1 (or in reality greater than something like 19930101) we are dealing with a rollback request to a specific date.

```
      \ifnum\pkgcls@targetdate>\@ne  % a real request
```

So we parse the date of this release to check if it is before or after the request date.

```
        \ifnum\@parse@version#2//00\@nil
             >\pkgcls@targetdate
```

If it is after we have to distinguish between two cases: If there was an earlier candidate we use that one because the other is too late, but if there wasn't one (i.e., if current release is the oldest that exists) we use it as the best choice. However in that case something is wrong (as there shouldn't be a rollback to a date where a package used doesn't yet exists. So we make a complained to the user.

```
          \ifx\pkgcls@candidate\@empty
            \pkgcls@rollbackdate@error{#2}%
            \pkgcls@use@this@release{#3}{#2}%
          \else
            \pkgcls@use@this@release\pkgcls@candidate
                                     \pkgcls@releasedate
          \fi
        \else
```

Otherwise, if the release date of this version is before the target rollback and we record it as a candidate. But we don't use it yet as there may be another release which is still before the target rollback.

```
          \def\pkgcls@candidate{#3}%
          \def\pkgcls@releasedate{#2}%
⟨*tracerollback⟩
          \pkgcls@debug{New candidate: #3}%
⟨/tracerollback⟩
        \fi
      \else
```

If we end up in this branch we have a named version request. So we check if `\pkgcls@targetlabel` matches the current name and if yes we use this release immediately, otherwise we do nothing as a later declaration may match it.

```
        \def\reserved@a{#1}%
        \ifx\pkgcls@targetlabel\reserved@a
          \pkgcls@use@this@release{#3}{#2}%
⟨*tracerollback⟩
        \else
          \pkgcls@debug{Label doesn't match}%
⟨/tracerollback⟩
        \fi
      \fi
    \fi
  \fi
}
```

(*End definition for* `\DeclareRelease`*. This function is documented on page* **??**.)

`\pkgcls@use@this@release` If a certain release has been selected (stored in the external file given in `#1`) we need to input it and afterwards stop reading the current file.

```
\def\pkgcls@use@this@release#1#2{%
```

Before that we record the selection made inside the transcript.

```
  \pkgcls@show@selection{#1}{#2}%
```

We then set the `\pkgcls@targetdate` to zero so that any `\DeclareRelease` or `\DeclareCurrentRelease` in the file we now load are bypassed[18] and then we finally load the correct release.

[18]The older release may also have such declarations inside if it was a simply copy of the `.sty` or `.cls` file current at that date. Removing these declarations would make the file load a tiny bit faster, but this way it works in any case.

After loading that file we need to stop reading the current file so we issue `\endinput`. Note that the `\relax` before that is essential to ensure that the `\endinput` is only happening after the file has been fully processed, otherwise it would act after the first line of the `\@@input`!

```
   \pkgcls@targetdate\z@
   \@@input #1\relax
   \endinput
}
```

(*End definition for* `\pkgcls@use@this@release`*. This function is documented on page* **??**.)

`\pkgcls@show@selection` This command records what selection was made. As that is needed in two places (and it is rather lengthly) it was placed in a separate command. The first argument is the name of the external file that is being loaded and is only needed for debugging. The second argument is the date that corresponds to this file and it is used as part of the message.

```
\def\pkgcls@show@selection#1#2{%
⟨*tracerollback⟩
  \pkgcls@debug{Result: use  #1}%
⟨/tracerollback⟩
  \GenericInfo
   {\@spaces\@spaces\space}{Rollback for
    \@cls@pkg\space'\@currname' requested ->
    \ifnum\pkgcls@targetdate>\@ne
       date
       \ifnum\requestedLaTeXdate=\pkgcls@targetdate
          \requestedpatchdate
       \else
          \expandafter\@gobble\pkgcls@arg
       \fi.\MessageBreak
```

Instead of "best approximation" we could say that we have been able to exactly match the date (if it is exact), but that would mean extra tests without much gain, so not done.

```
       Best approximation is
    \else
       version '\pkgcls@targetlabel'.\MessageBreak
       This corresponds to
    \fi
    \ifx\@nil#2\@nil
       a special release%
    \else
       the release introduced on #2%
    \fi
    \@gobble}%
}
```

(*End definition for* `\pkgcls@show@selection`*. This function is documented on page* **??**.)

`\pkgcls@rollbackdate@error` This is called if the requested rollback date is earlier than the earliest known release of a package or class.

A similar error is given if global rollback date and min-date on a specific package conflict with each other, but that case is happens only once so it is inlined.

```
\def\pkgcls@rollbackdate@error#1{%
  \@latex@error{Suspicious rollback date given}%
     {The \@cls@pkg\space'\@currname'  has no rollback data
```

```
      before #1 which\MessageBreak
      is after your requested rollback date --- so
      something may be wrong here.\MessageBreak
      Continue and we use the earliest known release.}}
```

(*End definition for* `\pkgcls@rollbackdate@error`. *This function is documented on page* ??.)

`\DeclareCurrentRelease` This declares the date (and possible name) of the current version of a package or class.

```
\def\DeclareCurrentRelease#1#2{%
```

First we test if `\pkgcls@targetdate` is greater than zero, otherwise this code is bypassed (as there is no rollback request).

```
  \ifnum\pkgcls@targetdate>\z@  % some sort of rollback request
⟨*tracerollback⟩
    \pkgcls@debug{---DeclareCurrentRelease}%
    \pkgcls@debug{   1: #1}%
    \pkgcls@debug{   2: #2}%
⟨/tracerollback⟩
```

If the value is greater than 1 we have to deal with a date request, so we parse `#2` as a date and compare it with `\pkgcls@targetdate`.

```
    \ifnum\pkgcls@targetdate>\@ne  % a date request
      \ifnum\@parse@version#2//00\@nil
           >\pkgcls@targetdate
```

If it is greater that means the release date if this file is later than the requested rollback date. Again we have two cases: If there was a previous candidate release we use that one as the current release is too young, but if there wasn't we have to use this release nevertheless as there isn't any alternative.

However this case can only happen if there is a `\DeclareCurrentRelease` but no declared older releases (so basically the use of the declaration is a bit dubious).

```
        \ifx\pkgcls@candidate\@empty
          \pkgcls@rollbackdate@error{#2}%
        \else
          \pkgcls@use@this@release\pkgcls@candidate
                                   \pkgcls@releasedate
        \fi
```

Otherwise the curent file is the right release, so we record that in the transcript and then carry on.

```
      \else
        \pkgcls@show@selection{current version}{#2}%
      \fi
    \else % a label request
```

Otherwise we have a rollback request to a named version so we check if that fits the current name and if not give an error as this was the last possible opportunity.

```
      \def\reserved@a{#1}%
      \ifx\pkgcls@targetlabel\reserved@a
        \pkgcls@show@selection{current version}{#2}%
      \else
        \@latex@error{Requested version '\pkgcls@targetlabel' for
          \@cls@pkg\space'\@currname' is unknown}\@ehc
      \fi
    \fi
```

```
  \fi
}
```

(*End definition for* `\DeclareCurrentRelease`*. This function is documented on page* **??**.)

`\IfTargetDateBefore` This enables a simple form of conditional code inside a class or package file. If there is a date request and the request date is earlier than the first argument the code in the second argument is processed otherwise the code in the third argument is processed. If there was no date request then we also execute the third argument, i.e., we will get the "latest" version of the file.

Most often the second argument (before-date-code) will be empty.

```
\DeclareRobustCommand\IfTargetDateBefore[1]{%
  \ifnum\pkgcls@innerdate <%
        \expandafter\@parse@version\expandafter0#1//00\@nil
    \typeout{Exclude code introduced on #1}%
    \expandafter\@firstoftwo
  \else
    \typeout{Include code introduced on #1}%
    \expandafter\@secondoftwo
  \fi
}
```

(*End definition for* `\IfTargetDateBefore`*. This function is documented on page* **??**.)

⟨/2ekernel | latexreleasefirst⟩

6 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}
         [2020-08-08 v1.3m 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 Q
ltfilehook.dtx

Contents

1 Introduction

1.1 Provided hooks

The code offers a number of hooks into which packages (or the user) can add code to support different use cases. Many hooks are offered as pairs (i.e., the second hook is reversed. Also important to know is that these pairs are properly nested with respect to other pairs of hooks.

There are hooks that are executed for all files of a certain type (if they contain code), e.g., for all "include files" or all "packages", and there are also hooks that are specific to a single file, e.g., do something after the package `foo.sty` has been loaded.

1.2 General hooks for file reading

There are four hooks that are called for each file that is read using document-level commands such as `\input`, `\include`, `\usepackage`, etc. They are not called for files read using internal low-level methods, such as `\@input` or `\openin`.

`file/before`
`file/before/...`
`file/after/...`
`file/after`

These are:

`file/before`, `file/before/`⟨*file-name*⟩ These hooks are executed in that order just before the file is loaded for reading. The code of the first hook is used with every file, while the second is executed only for the file with matching ⟨*file-name*⟩ allowing you to specify code that only applies to one file.

`file/after/`⟨*file-name*⟩, `file/after` These hooks are after the file with name ⟨*file-name*⟩ has been fully consumed. The order is swapped (the specific one comes first) so that the `before` and `after` hooks nest properly, which is important if any of them involve grouping (e.g., contain environments, for example). Furthermore both hooks are reversed hooks to support correct nesting of different packages adding code to both `/before` and `/after` hooks.

So the overall sequence of hook processing for any file read through the user interface commands of LaTeX is:

```
\UseHook{⟨file/before⟩}
\UseHook{⟨file/before/⟨file name⟩⟩}
    ⟨file contents⟩
\UseHook{⟨file/after/⟨file name⟩⟩}
\UseHook{⟨file/after⟩}
```

The file hooks only refer to the file by its name and extension, so the ⟨*file name*⟩ should be the file name as it is on the filesystem with extension (if any) and without paths. Different from `\input` and similar commands, the `.tex` extension is not assumed in hook ⟨*file name*⟩, so `.tex` files must be specified with their extension to be recognized. Files within subfolders should also be addressed by their name and extension only.

Extensionless files also work, and should then be given without extension. Note however that TEX prioritizes `.tex` files, so if two files `foo` and `foo.tex` exist in the search path, only the latter will be seen.

When a file is input, the ⟨*file name*⟩ is available in `\CurrentFile`, which is then used when accessing the `file/before/`⟨*file name*⟩ and `file/after/`⟨*file name*⟩.

`\CurrentFile`

The name of the file about to be read (or just finished) is available to the hooks through `\CurrentFile` (there is no `expl3` name for it for now). The file is always provided with its extension, i.e., how it appears on your hard drive, but without any specified path to it. For example, `\input{sample}` and `\input{app/sample.tex}` would both have `\CurrentFile` being `sample.tex`.

`\CurrentFilePath`

The path to the current file (complement to `\CurrentFile`) is available in `\CurrentFilePath` if needed. The paths returned in `\CurrentFilePath` are only user paths, given through `\input@path` (or expl3's equivalent `\l_file_search_path_seq`) or by directly typing in the path in the `\input` command or equivalent. Files located by `kpsewhich` get the path added internally by the TEX implementation, so at the macro level it looks as if the file were in the current folder, so the path in `\CurrentFilePath` is empty in these cases (package and class files, mostly).

`\CurrentFileUsed`
`\CurrentFilePathUsed`

In normal circumstances these are identical to `\CurrentFile` and `\CurrentFilePath`. They will differ when a file substitution has occurred for `\CurrentFile`. In that case, `\CurrentFileUsed` and `\CurrentFilePathUsed` will hold the actual file name and path loaded by LATEX, while `\CurrentFile` and `\CurrentFilePath` will hold the names that were *asked for*. Unless doing very specific work on the file being read, `\CurrentFile` and `\CurrentFilePath` should be enough.

1.3 Hooks for package and class files

Commands to load package and class files (e.g., `\usepackage`, `\RequirePackage`, `\LoadPackageWithOptions`, etc.) offer the hooks from section 1.2 when they are used to load a package or class file, e.g., `file/after/array.sty` would be called after the array package got loaded. But as packages and classes form as special group of files, there are some additional hooks available that only apply when a package or class is loaded.

package/before
package/after
package/before/...
package/after/...
class/before
class/after
class/before/...
class/after/...

These are:

package/before, package/after These hooks are called for each package being loaded.

package/before/⟨*name*⟩, package/after/⟨*name*⟩ These hooks are additionally called if the package name is ⟨*name*⟩ (without extension).

class/before, class/after These hooks are called for each class being loaded.

class/before/⟨*name*⟩, class/after/⟨*name*⟩ These hooks are additionally called if the class name is ⟨*name*⟩ (without extension).

All `/after` hooks are implemented as reversed hooks.
The overall sequence of execution for `\usepackage` and friends is therefore:

```
\UseHook{⟨package/before⟩}
\UseHook{⟨package/before/⟨package name⟩⟩}
    \UseHook{⟨file/before⟩}
    \UseHook{⟨file/before/⟨package name⟩.sty⟩}
        ⟨package contents⟩
    \UseHook{⟨file/after/⟨package name⟩.sty⟩}
    \UseHook{⟨file/after⟩}
    code from \AtEndOfPackage if used inside the package
\UseHook{⟨package/after/⟨package name⟩⟩}
\UseHook{⟨package/after⟩}
```

and similar for class file loading, except that `package/` is replaced by `class/` and `\AtEndOfPackage` by `\AtEndOfClass`.

If a package or class is not loaded (or it was loaded before the hooks were set) none of the hooks are executed!

1.4 Hooks for `\include` files

To manage `\include` files, LaTeX issues a `\clearpage` before and after loading such a file. Depending on the use case one may want to execute code before or after these `\clearpage`s especially for the one that is issued at the end.

Executing code before the final `\clearpage`, means that the code is processed while the last page of the included material is still under construction. Executing code after it means that all floats from inside the include file are placed (which might have added further pages) and the final page has finished.

Because of these different scenarios we offer hooks in three places.[19] None of the hooks are executed when an `\include` file is bypassed because of an `\includeonly` declaration. They are, however, all executed if LaTeX makes an attempt to load the `\include` file (even if it doesn't exist and all that happens is "`No file ⟨filename⟩.tex`").

[19] If you want to execute code before the first `\clearpage` there is no need to use a hook—you can write it directly in front of the `\include`.

include/before
include/before/...
include/end
include/end/...
include/after
include/after/...

These are:

include/before, include/before/⟨*name*⟩ These hooks are executed (in that order) after the initial `\clearpage` and after `.aux` file is changed to use ⟨*name*⟩`.aux`, but before the ⟨*name*⟩`.tex` file is loaded. In other words they are executed at the very beginning of the first page of the `\include` file.

include/end/⟨*name*⟩, include/end These hooks are executed (in that order) after LaTeX has stopped reading from the `\include` file, but before it has issued a `\clearpage` to output any deferred floats.

include/after/⟨*name*⟩, include/after These hooks are executed (in that order) after LaTeX has issued the `\clearpage` but before is has switched back writing to the main `.aux` file. Thus technically we are still inside the `\include` and if the hooks generate any further typeset material including anything that writes to the `.aux` file, then it would be considered part of the included material and bypassed if it is not loaded because of some `\includeonly` statement.[20]

1.5 High-level interfaces for LaTeX

We do not provide any high-level LaTeX commands (like filehook or scrlfile do) but think that for package writers the commands from for hook management are sufficient.

1.6 Internal interfaces for LaTeX

\declare@file@substitution
\undeclare@file@substitution

`\declare@file@substitution {⟨file⟩} {⟨replacement-file⟩}`
`\undeclare@file@substitution {⟨file⟩}`

If ⟨*file*⟩ is requested for loading replace it with ⟨*replacement-file*⟩. `\CurrentFile` remains pointing to ⟨*file*⟩ but `\CurrentFileUsed` will show the file actually loaded.

The main use case for this declaration is to provide a corrected version of a package that can't be changed (due to its license) but no longer functions because of LaTeX kernel changes, for example, or to provide a version that makes use of new kernel functionality while the original package remains available for use with older releases.

The `\undeclare@file@substitution` declaration undoes a substitution made earlier.

> *Please do not misuse this functionality and replace a file with another unless if really needed and only if the new version is implementating the same functionality as the original one!*

[20] For that reason another `\clearpage` is executed after these hooks which normally does nothing, but starts a new page if further material got added this way.

\disable@package@load
\reenable@package@load

\disable@package@load {⟨*package*⟩} {⟨*alternate-code*⟩}
\reenable@package@load {⟨*package*⟩}

If ⟨*package*⟩ is requested do not load it but instead run ⟨*alternate-code*⟩ which could issue a warning, error or any other code.

The main use case is for classes that want to restrict the set of supported packages or contain code that make the use of some packages impossible. So rather than waiting until the document breaks they can set up informative messages why certain packages are not available.

The function is only implemented for packages not for arbitrary files.

1.7 A sample package for structuring the log output

As an application we provide the package structuredlog that adds lines to the `.log` when a file is opened and closed for reading keeping track of nesting level es well. For example, for the current document it adds the lines

```
= (LEVEL 1 START) t1lmr.fd
= (LEVEL 1 STOP) t1lmr.fd
= (LEVEL 1 START) supp-pdf.mkii
= (LEVEL 1 STOP) supp-pdf.mkii
= (LEVEL 1 START) nameref.sty
== (LEVEL 2 START) refcount.sty
== (LEVEL 2 STOP) refcount.sty
== (LEVEL 2 START) gettitlestring.sty
== (LEVEL 2 STOP) gettitlestring.sty
= (LEVEL 1 STOP) nameref.sty
= (LEVEL 1 START) ltfilehook-doc.out
= (LEVEL 1 STOP) ltfilehook-doc.out
= (LEVEL 1 START) ltfilehook-doc.out
= (LEVEL 1 STOP) ltfilehook-doc.out
= (LEVEL 1 START) ltfilehook-doc.hd
= (LEVEL 1 STOP) ltfilehook-doc.hd
= (LEVEL 1 START) ltfilehook.dtx
== (LEVEL 2 START) ot1lmr.fd
== (LEVEL 2 STOP) ot1lmr.fd
== (LEVEL 2 START) omllmm.fd
== (LEVEL 2 STOP) omllmm.fd
== (LEVEL 2 START) omslmsy.fd
== (LEVEL 2 STOP) omslmsy.fd
== (LEVEL 2 START) omxlmex.fd
== (LEVEL 2 STOP) omxlmex.fd
== (LEVEL 2 START) umsa.fd
== (LEVEL 2 STOP) umsa.fd
== (LEVEL 2 START) umsb.fd
== (LEVEL 2 STOP) umsb.fd
== (LEVEL 2 START) ts1lmr.fd
== (LEVEL 2 STOP) ts1lmr.fd
== (LEVEL 2 START) t1lmss.fd
== (LEVEL 2 STOP) t1lmss.fd
= (LEVEL 1 STOP) ltfilehook.dtx
```

Thus if you inspect an issue in the `.log` it is easy to figure out in which file it occurred, simply by searching back for `LEVEL` and if it is a `STOP` then remove 1 from the level value and search further for `LEVEL` with that value which should then be the `START` level of the file you are in.

2 The Implementation

```
⟨*2ekernel⟩
⟨@@=filehook⟩
```

2.1 Document and package-level commands

\CurrentFile
\CurrentFilePath
\CurrentFileUsed
\CurrentFilePathUsed

User-level macros that hold the current file name and file path. These are used internally as well because the code takes care to protect against a possible redefinition of these macros in the loaded file (it's necessary anyway to make hooks work with nested `\input`). The versions `\...Used` hold the *actual* file name and path that is loaded by LaTeX, whereas the other two hold the name as requested. They will differ in case there's a file substitution.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\CurrentFile}{Hook management file}%
\ExplSyntaxOn
\tl_new:N \CurrentFile
\tl_new:N \CurrentFilePath
\tl_new:N \CurrentFileUsed
\tl_new:N \CurrentFilePathUsed
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\CurrentFile}{Hook management file}%
⟨latexrelease⟩
⟨latexrelease⟩\let \CurrentFile         \@undefined
⟨latexrelease⟩\let \CurrentFilePath     \@undefined
⟨latexrelease⟩\let \CurrentFileUsed     \@undefined
⟨latexrelease⟩\let \CurrentFilePathUsed \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\CurrentFile` *and others. These functions are documented on page* 680.)

2.2 expl3 helpers

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\__filehook_file_parse_full_name:nN}{File helpers}%
\ExplSyntaxOn
```

__filehook_file_parse_full_name:nN
__filehook_full_name:nn
__filehook_set_curr_file_assign:nnnNN

A utility macro to trigger expl3's file-parsing and lookup, and return a normalized representation of the file name. If the queried file doesn't exist, no normalisation takes place.

The output of __filehook_file_parse_full_name:nN is passed on to the #2—a 3-argument macro that takes the ⟨*path*⟩, ⟨*base*⟩, and ⟨*ext*⟩ parts of the file name.

```
\cs_new:Npn \__filehook_file_parse_full_name:nN #1
  {
    \exp_args:Nf \file_parse_full_name_apply:nN
      {
        \exp_args:Nf \__filehook_full_name:nn
          { \file_full_name:n {#1} } {#1}
      }
  }
\cs_new:Npn \__filehook_full_name:nn #1 #2
  {
    \tl_if_empty:nTF {#1}
      { \tl_trim_spaces:n {#2} }
      { \tl_trim_spaces:n {#1} }
  }
```

(*End definition for* __filehook_file_parse_full_name:nN*,* __filehook_full_name:nn*, and* __filehook_set_curr_file_assign:nnnNN.)

__filehook_if_no_extension:nTF
__filehook_drop_extension:N

Some actions depend on whether the file extension was explicitly given, and sometimes the extension has to be removed. The macros below use __filehook_file_parse_full_name:nN to split up the file name and either check if ⟨*ext*⟩ (#3) is empty, or discard it.

```
\cs_new:Npn \__filehook_if_no_extension:nTF #1
  {
    \exp_args:Ne \tl_if_empty:nTF
      { \file_parse_full_name_apply:nN {#1} \use_iii:nnn }
  }
\cs_new_protected:Npn \__filehook_drop_extension:N #1
  {
    \tl_gset:Nx #1
      {
        \exp_args:NV \__filehook_file_parse_full_name:nN #1
          \__filehook_drop_extension_aux:nnn
      }
  }
\cs_new:Npn \__filehook_drop_extension_aux:nnn #1 #2 #3
   { \tl_if_empty:nF {#1} { #1 / } #2 }
```

(*End definition for* __filehook_if_no_extension:nTF *and* __filehook_drop_extension:N.)

\g__filehook_input_file_seq
\l__filehook_internal_tl
__filehook_file_push:
__filehook_file_pop:
__filehook_file_pop_assign:nnnn

Yet another stack, to keep track of \CurrentFile and \CurrentFilePath with nested \inputs. At the beginning of \InputIfFileExists, the current value of \CurrentFilePath and \CurrentFile is pushed to \g__filehook_input_file_seq, and at the end, it is popped and the value reassigned. Some other places don't use \InputIfFileExists directly (\include) or need \CurrentFile earlier (\@onefilewithoptions), so these are manually used elsewhere as well.

```
\tl_new:N \l__filehook_internal_tl
\seq_new:N \g__filehook_input_file_seq
\cs_new_protected:Npn \__filehook_file_push:
  {
    \seq_gpush:Nx \g__filehook_input_file_seq
```

```
      {
        { \CurrentFilePathUsed } { \CurrentFileUsed }
        { \CurrentFilePath     } { \CurrentFile     }
      }
  }
\cs_new_protected:Npn \__filehook_file_pop:
  {
    \seq_gpop:NNTF \g__filehook_input_file_seq \l__filehook_internal_tl
      { \exp_after:wN \__filehook_file_pop_assign:nnnn \l__filehook_internal_tl }
      {
        \msg_error:nnn { hooks } { should-not-happen }
          { Tried~to~pop~from~an~empty~file~name~stack. }
      }
  }
\cs_new_protected:Npn \__filehook_file_pop_assign:nnnn #1 #2 #3 #4
  {
    \tl_set:Nn \CurrentFilePathUsed {#1}
    \tl_set:Nn \CurrentFileUsed {#2}
    \tl_set:Nn \CurrentFilePath {#3}
    \tl_set:Nn \CurrentFile {#4}
  }
\ExplSyntaxOff
```

(*End definition for* `\g__filehook_input_file_seq` *and others.*)

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

When rolling forward the following expl3 functions may not be defined. If we roll back the code does nothing.

```
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                 {\file_parse_full_name_apply:nN}{Roll forward help}%
⟨latexrelease⟩
⟨latexrelease⟩\ExplSyntaxOn
⟨latexrelease⟩\cs_if_exist:NF\file_parse_full_name_apply:nN
⟨latexrelease⟩{
⟨latexrelease⟩\cs_new:Npn \file_parse_full_name_apply:nN #1
⟨latexrelease⟩  {
⟨latexrelease⟩    \exp_args:Ne \__file_parse_full_name_auxi:nN
⟨latexrelease⟩      { \__kernel_file_name_sanitize:n {#1} }
⟨latexrelease⟩  }
⟨latexrelease⟩\cs_new:Npn \__file_parse_full_name_auxi:nN #1
⟨latexrelease⟩  {
⟨latexrelease⟩    \__file_parse_full_name_area:nw { } #1
⟨latexrelease⟩      / \s__file_stop
⟨latexrelease⟩  }
⟨latexrelease⟩\cs_new:Npn \__file_parse_full_name_area:nw #1 #2 / #3 \s__file_stop
⟨latexrelease⟩  {
⟨latexrelease⟩    \tl_if_empty:nTF {#3}
⟨latexrelease⟩      { \__file_parse_full_name_base:nw { } #2 . \s__file_stop {#1} }
⟨latexrelease⟩      { \__file_parse_full_name_area:nw { #1 / #2 } #3 \s__file_stop }
⟨latexrelease⟩  }
⟨latexrelease⟩\cs_new:Npn \__file_parse_full_name_base:nw #1 #2 . #3 \s__file_stop
⟨latexrelease⟩  {
⟨latexrelease⟩    \tl_if_empty:nTF {#3}
```

```
⟨latexrelease⟩      {
⟨latexrelease⟩        \tl_if_empty:nTF {#1}
⟨latexrelease⟩          {
⟨latexrelease⟩            \tl_if_empty:nTF {#2}
⟨latexrelease⟩              { \__file_parse_full_name_tidy:nnnN { } { } }
⟨latexrelease⟩              { \__file_parse_full_name_tidy:nnnN { .#2 } { } }
⟨latexrelease⟩          }
⟨latexrelease⟩          { \__file_parse_full_name_tidy:nnnN {#1} { .#2 } }
⟨latexrelease⟩      }
⟨latexrelease⟩      { \__file_parse_full_name_base:nw { #1 . #2 } #3 \s__file_stop }
⟨latexrelease⟩  }
⟨latexrelease⟩\cs_new:Npn \__file_parse_full_name_tidy:nnnN #1 #2 #3 #4
⟨latexrelease⟩  {
⟨latexrelease⟩    \exp_args:Nee #4
⟨latexrelease⟩      {
⟨latexrelease⟩        \str_if_eq:nnF {#3} { / } { \use_none:n }
⟨latexrelease⟩        #3 \prg_do_nothing:
⟨latexrelease⟩      }
⟨latexrelease⟩      { \use_none:n #1 \prg_do_nothing: }
⟨latexrelease⟩      {#2}
⟨latexrelease⟩  }
⟨latexrelease⟩}
⟨latexrelease⟩\ExplSyntaxOff
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
⟨@@=⟩
```

2.3 Declaring the file-related hooks

All hooks starting with `file/ include/`, `class/` or `package/` are generic and will be allocated if code is added to them. Thus there is no need to explicitly declare any hook in the code below.

Furthermore, those named `.../after` or `.../end` are automatically declared as reversed hooks if filled with code, so this is also automatically taken care of.

2.4 Patching LaTeX's `\InputIfFileExists` command

Most of what we have to do is adding `\UseHook` into several LaTeX 2_ε core commands, because of some circular dependencies in the kernel we do this only now and not in `ltfiles`.

`\InputIfFileExists` `\@input@file@exists@with@hooks` `\unqu@tefilef@und`

`\InputIfFileExists` loads any file if it is available so we have to add the hooks `file/before` and `file/after` in the right places. If the file doesn't exist no hooks should be executed.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\InputIfFileExists}{Hook management (files)}%
⟨*2ekernel | latexrelease⟩
```

```
\let\InputIfFileExists\@undefined
\DeclareRobustCommand \InputIfFileExists[2]{%
  \IfFileExists{#1}%
```

```
    {%
     \@expl@@@filehook@file@push@@
     \@filehook@set@CurrentFile
```

If the file exists then \CurrentFile holds its name. But we can't rely on that still being true after the file has been processed. Thus for using the name in the file hooks we need to preserve the name and then restore it for the file/after/... hook.

The hook always refers to the file requested by the user. The hook is *always* loaded for \CurrentFile which usually is the same as \CurrentFileUsed. In the case of a file replacement, the \CurrentFileUsed holds the actual file loaded. In any case the file names are normalized so that the hooks work on the real file name, rather than what the user typed in.

expl3's \file_full_name:n normalizes the file name (to factor out differences in the .tex extension), and then does a file lookup to take into account a possible path from \l_file_search_path_seq and \input@path. However only the file name and extension are returned so that file hooks can refer to the file by their name only. The path to the file is returned in \CurrentFilePath.

```
      \edef\reserved@a{%
        \@expl@@@filehook@file@pop@assign@@nnnn
          {\CurrentFilePathUsed}%
          {\CurrentFileUsed}%
          {\CurrentFilePath}%
          {\CurrentFile}%
```

We pre-expand \@filef@und so that in case another file is loaded in the true branch of \InputIfFileExists, these don't change their value meanwhile. This isn't a worry with \CurrentFile... because they are kept in a stack.

```
        \noexpand\@input@file@exists@with@hooks{\@filef@und}}%
      \expandafter\@swaptwoargs\expandafter
        {\reserved@a}%
        {#2}%
      \@expl@@@filehook@file@pop@@
    }%
}
```

Before adding to the file list we need to make all (letter) characters catcode 11, because several packages use constructions like

```
\filename@parse{<filename>}
\ifx\filename@ext\@clsextension
  ...
\fi
```

and that doesn't work if \filename@ext is \detokenized. Making \@clsextension a string doesn't help much because some packages define their own \<prefix>@someextension with normal catcodes. This is not entirely correct because packages loaded (somehow) with catcode 12 alphabetic tokens (say, as the result of a \string or \detokenize command, or from a TEX string like \jobname) will have these character tokens incorrectly turned into letter tokens. This however is rare, so we'll go for the all-letters approach (grepping the packages in TEX Live didn't bring up any obvious candidate for breaking with this catcode change).

```
\def\@input@file@exists@with@hooks#1{%
  \edef\reserved@a{\unqu@tefilef@und#1\@nil}%
```

```
  \@addtofilelist{\string@makeletter\reserved@a}%
  \UseHook{file/before}%
```

The current file name is available in `\CurrentFile` so we use that in the specific hook.

```
  \UseHook{file/before/\CurrentFile}%
  \@@input #1% <- trailing space comes from \@filef@und
```

And it is restored here so we can use it once more.

```
  \UseHook{file/after/\CurrentFile}%
  \UseHook{file/after}}
\def\unqu@tefilef@und"#1" \@nil{#1}
⟨latexrelease⟩\EndIncludeInRelease
⟨/2ekernel | latexrelease⟩
```

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`'.

```
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩          {\InputIfFileExists}{Hook management (files)}%
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand \InputIfFileExists[2]{%
⟨latexrelease⟩  \IfFileExists{#1}%
⟨latexrelease⟩    {%
⟨latexrelease⟩  \expandafter\@swaptwoargs\expandafter
⟨latexrelease⟩      {\@filef@und}{#2\@addtofilelist{#1}\@@input}}}
⟨latexrelease⟩\let\@input@file@exists@with@hooks\@undefined
⟨latexrelease⟩\let\unqu@tefilef@und\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩          {\InputIfFileExists}{Hook management (files)}%
⟨latexrelease⟩\long\def \InputIfFileExists#1#2{%
⟨latexrelease⟩  \IfFileExists{#1}%
⟨latexrelease⟩    {#2\@addtofilelist{#1}\@@input \@filef@und}}
⟨latexrelease⟩\let\@input@file@exists@with@hooks\@undefined
⟨latexrelease⟩\let\unqu@tefilef@und\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\InputIfFileExists`, `\@input@file@exists@with@hooks`, *and* `\unqu@tefilef@und`. *These functions are documented on page* ??.)

2.5 Declaring a file substitution

```
⟨@@=filehook⟩
```

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\__filehook_subst_add:nn}{Declaring file substitution}%
\ExplSyntaxOn
```

`\__filehook_subst_add:nn`
`\__filehook_subst_remove:n`
`\__filehook_subst_file_normalize:Nn`
`\__filehook_subst_empty_name_chk:NN`

`\__filehook_subst_add:nn` declares a file substitution by doing a (global) definition of the form `\def\@file-subst@`⟨*file*⟩`{`⟨*replacement*⟩`}`. The file names are properly sanitised, and normalized with the same treatment done for the file hooks. That is, a file replacement is declared by using the file name (and extension, if any) only, and the file path should not be given. If a file name is empty it is replaced by `.tex` (the empty csname is used to check that).

```
\cs_new_protected:Npn \__filehook_subst_add:nn #1 #2
  {
    \group_begin:
      \cs_set:cpx { } { \exp_not:o { \cs:w\cs_end: } }
      \int_set:Nn \tex_escapechar:D { -1 }
      \cs_gset:cpx
        {
          @file-subst@
          \__filehook_subst_file_normalize:Nn \use_ii_iii:nnn {#1}
        }
        { \__filehook_subst_file_normalize:Nn \__filehook_file_name_compose:nnn {#2} }
    \group_end:
  }
\cs_new_protected:Npn \__filehook_subst_remove:n #1
  {
    \group_begin:
      \cs_set:cpx { } { \exp_not:o { \cs:w\cs_end: } }
      \int_set:Nn \tex_escapechar:D { -1 }
      \cs_undefine:c
        {
          @file-subst@
          \__filehook_subst_file_normalize:Nn \use_ii_iii:nnn {#1}
        }
    \group_end:
  }
\cs_new:Npn \__filehook_subst_file_normalize:Nn #1 #2
  {
    \exp_after:wN \__filehook_subst_empty_name_chk:NN
      \cs:w \exp_after:wN \cs_end:
        \cs:w \__filehook_file_parse_full_name:nN {#2} #1 \cs_end:
  }
\cs_new:Npn \__filehook_subst_empty_name_chk:NN #1 #2
  { \if_meaning:w #1 #2 .tex \else: \token_to_str:N #2 \fi: }
```

(*End definition for* `\__filehook_subst_add:nn` *and others.*)

`\use_ii_iii:nnn` A variant of `\use_...` to discard the first of three arguments.

Todo: this should move to expl3

```
\cs_gset:Npn \use_ii_iii:nnn #1 #2 #3 {#2 #3}
```

(*End definition for* `\use_ii_iii:nnn`.)

```
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

`\declare@file@substitution` `\undeclare@file@substitution` For two internals we provide LaTeX 2ε names so that we can use them elsewhere in the kernel (and so that they can be used in packages if really needed, e.g., scrlfile).

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\declare@file@substitution}{File substitution}%
\ExplSyntaxOn
```

```
\cs_new_eq:NN \declare@file@substitution   \__filehook_subst_add:nn
\cs_new_eq:NN \undeclare@file@substitution \__filehook_subst_remove:n
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩          {\declare@file@substitution}{File substitution}%
⟨latexrelease⟩
⟨latexrelease⟩\let \declare@file@substitution   \@undefined
⟨latexrelease⟩\let \undeclare@file@substitution \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\declare@file@substitution` *and* `\undeclare@file@substitution`*. These functions are documented on page 682.*)

```
⟨@@=⟩
\ExplSyntaxOff
```

2.6 Selecting a file (`\set@curr@file`)

`\set@curr@file`
`\@curr@file`
`\@curr@file@reqd`

Now we hook into `\set@curr@file` to resolve a possible file substitution, and add `\@expl@@@filehook@set@curr@file@@nNN` at the end, after `\@curr@file` is set.

A file name is built using `\expandafter\string\csname`⟨*filename*⟩`\endcsname` to avoid expanding utf8 active characters. The `\csname` expands the normalisation machinery and the routine to resolve a file substitution, returning a control sequence with the same name as the file.

It happens that when ⟨*filename*⟩ is empty, the generated control sequence is `\csname\endcsname`, and doing `\string` on that results in the file `csnameendcsname.tex`. To guard against that we `\ifx`-compare the generated control sequence with the empty csname. To do so, `\csname\endcsname` has to be defined, otherwise it would be equal to `\relax` and we would have false positives. Here we define `\csname\endcsname` to expand to itself to avoid it matching the definition of some other control sequence.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\set@curr@file}{Setting current file name}%
\def\set@curr@file#1{%
  \begingroup
    \escapechar\m@ne
    \expandafter\def\csname\expandafter\endcsname
      \expandafter{\csname\endcsname}%
```

Two file names are set here: `\@curr@file@reqd` which is the file requested by the user, and `\@curr@file` which should be the same, except when we have a file substitution, in which case it holds the actual loaded file. `\@curr@file` is resolved first, to check if a substitution happens. If it doesn't, `\@expl@@@filehook@if@file@replaced@@TF` short-cuts and just copies `\@curr@file`, otherwise the full normalisation procedure is executed.

At this stage the file name is parsed and normalized, but if the input doesn't have an extension, the default `.tex` is *not* added to `\@curr@file` because for applications other than `\input` (graphics, for example) the default extension may not be

.tex. First check if the input has an extension, then if the input had no extension, call \@expl@@@filehook@drop@extension@@N. In case of a file substitution, \@curr@file will have an extension.

```
    \@expl@@@filehook@if@no@extension@@nTF{#1}%
      {\@tempswatrue}{\@tempswafalse}%
    \@kernel@make@file@csname\@curr@file
      \@expl@@@filehook@resolve@file@subst@@w {#1}%
    \@expl@@@filehook@if@file@replaced@@TF
      {\@kernel@make@file@csname\@curr@file@reqd
         \@expl@@@filehook@normalize@file@name@@w{#1}%
       \if@tempswa \@expl@@@filehook@drop@extension@@N\@curr@file@reqd \fi}%
      {\if@tempswa \@expl@@@filehook@drop@extension@@N\@curr@file \fi
       \global\let\@curr@file@reqd\@curr@file}%
    \@expl@@@filehook@clear@replacement@flag@@
  \endgroup}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{2019/10/01}%
⟨latexrelease⟩          {\set@curr@file}{Setting current file name}%
⟨latexrelease⟩\def\set@curr@file#1{%
⟨latexrelease⟩  \begingroup
⟨latexrelease⟩    \escapechar\m@ne
⟨latexrelease⟩    \xdef\@curr@file{%
⟨latexrelease⟩      \expandafter\expandafter\expandafter\unquote@name
⟨latexrelease⟩      \expandafter\expandafter\expandafter{%
⟨latexrelease⟩      \expandafter\string
⟨latexrelease⟩        \csname\@firstofone#1\@empty\endcsname}}%
⟨latexrelease⟩  \endgroup
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩          {\set@curr@file}{Setting current file name}%
⟨latexrelease⟩\let\set@curr@file\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \set@curr@file*,* \@curr@file*, and* \@curr@file@reqd*. These functions are documented on page* ??.)

\@filehook@set@CurrentFile
\@kernel@make@file@csname
\@set@curr@file@aux

Todo: This should get internalized using @expl@ *names*

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\@kernel@make@file@csname}{Make file csname}%
```

```
\def\@kernel@make@file@csname#1#2#3{%
  \xdef#1{\expandafter\@set@curr@file@aux
    \csname\expandafter#2\@firstofone#3\@nil\endcsname}}
```

This auxiliary compares \⟨*filename*⟩ with \csname\endcsname to check if the empty .tex file was requested.

```
\def\@set@curr@file@aux#1{%
  \expandafter\ifx\csname\endcsname#1%
    .tex\else\string#1\fi}
```

Then we call \@expl@@@filehook@set@curr@file@@nNN once for \@curr@file to set \CurrentFile(Path)Used and once for \@curr@file@reqd to set \CurrentFile(Path). Here too the slower route is only used if a substitution happened, but here \@expl@@@filehook@if@file@ can't be used because the flag is reset at the \endgroup above, so we check if \@curr@file and \@curr@file@reqd differ. This macro is issued separate from \set@curr@file because it changes \CurrentFile, and side-effects would quickly get out of control.

```
\def\@filehook@set@CurrentFile{%
  \@expl@@@filehook@set@curr@file@@nNN{\@curr@file}%
    \CurrentFileUsed\CurrentFilePathUsed
  \ifx\@curr@file@reqd\@curr@file
    \let\CurrentFile\CurrentFileUsed
    \let\CurrentFilePath\CurrentFilePathUsed
  \else
    \@expl@@@filehook@set@curr@file@@nNN{\@curr@file@reqd}%
      \CurrentFile\CurrentFilePath
  \fi}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@filehook@set@CurrentFile*,* \@kernel@make@file@csname*, and* \@set@curr@file@aux*. These functions are documented on page* ??.)

\@@_set_curr_file:nNN
\@@_set_curr_file_assign:nnnNN

When inputting a file, \set@curr@file does a file lookup (in \input@path and \l_file_search_path_seq) and returns the actual file name (⟨*base*⟩ plus ⟨*ext*⟩) in \CurrentFileUsed, and in case there's a file substitution, the requested file in \CurrentFile (otherwise both are the same). Only the base and extension are returned, regardless of the input (both path/to/file.tex and file.tex end up as file.tex in \CurrentFile). The path is returned in \CurrentFilePath, in case it's needed.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {@@_set_curr_file:nNN}{Set curr file}%
\ExplSyntaxOn
⟨@@=filehook⟩
\cs_new_protected:Npn \__filehook_set_curr_file:nNN #1
  {
    \exp_args:Nf \__filehook_file_parse_full_name:nN {#1}
      \__filehook_set_curr_file_assign:nnnNN
  }
\cs_new_protected:Npn \__filehook_set_curr_file_assign:nnnNN #1 #2 #3 #4 #5
  {
    \str_set:Nn #5 {#1}
    \str_set:Nn #4 {#2#3}
  }
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* \@@_set_curr_file:nNN *and* \@@_set_curr_file_assign:nnnNN*. These functions are documented on page* ??.)

2.7 Replacing a file and detecting loops

__filehook_resolve_file_subst:w
__filehook_normalize_file_name:w
__filehook_file_name_compose:nnn

Start by sanitising the file with `\__filehook_file_parse_full_name:nN` then do `\__filehook_file_subst_begin:nnn`{⟨*path*⟩}{⟨*name*⟩}{⟨*ext*⟩}.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\__filehook_resolve_file_subst:w}{Replace files detect loops}%
\ExplSyntaxOn
\cs_new:Npn \__filehook_resolve_file_subst:w #1 \@nil
  { \__filehook_file_parse_full_name:nN {#1} \__filehook_file_subst_begin:nnn }
\cs_new:Npn \__filehook_normalize_file_name:w #1 \@nil
  { \__filehook_file_parse_full_name:nN {#1} \__filehook_file_name_compose:nnn }
\cs_new:Npn \__filehook_file_name_compose:nnn #1 #2 #3
  { \tl_if_empty:nF {#1} { #1 / } #2#3 }
```

flag␣__filehook_file_replaced
__filehook_if_file_replaced:TF
__filehook_clear_replacement_flag:

Since the file replacement is done expandably in a `\csname`, use a flag to remember if a substitution happened. We use this in `\set@curr@file` to short-circuit some of it in case no substitution happened (by far the most common case, so it's worth optimising). The flag raised during the file substitution algorithm must be explicitly cleared after the `\__filehook_if_file_replaced:TF` conditional is no longer needed, otherwise further uses of `\__filehook_if_file_replaced:TF` will wrongly return true.

```
\flag_new:n { __filehook_file_replaced }
\cs_new:Npn \__filehook_if_file_replaced:TF #1 #2
  { \flag_if_raised:nTF { __filehook_file_replaced } {#1} {#2} }
\cs_new_protected:Npn \__filehook_clear_replacement_flag:
  { \flag_clear:n { __filehook_file_replaced } }
```

__filehook_file_subst_begin:nnn

First off, start by checking if the current file (⟨*name*⟩ + ⟨*ext*⟩) has a declared substitution. If not, then just put that as the name (including a possible ⟨*path*⟩ in this case): this is the default case with no substitutions, so it's the first to be checked. The auxiliary `\__filehook_file_subst_tortoise_hare:nn` sees that there's no replacement for `#2#3` and does nothing else.

```
\cs_new:Npn \__filehook_file_subst_begin:nnn #1 #2 #3
  {
    \__filehook_file_subst_tortoise_hare:nn { #2#3 } { #2#3 }
      { \__filehook_file_name_compose:nnn {#1} {#2} {#3} }
  }
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

2.7.1 The Tortoise and Hare algorithm

__filehook_file_subst_tortoise_hare:nn
__filehook_file_subst_loop:NN
__filehook_file_subst_loop:cc

If there is a substitution (⟨*true*⟩ in the first `\cs_if_exist:cTF` below), then first check if there is no substitution down the line: this should be the second most common case, of one file replaced by another. In that case just leave the substitution there and the job is done. If any substitution happens, then the `\flag __filehook_file_replaced` is raised (conditionally, because checking if a flag is raised is much faster than raising it over and over again).

If, however there are more substitutions, then we need to check for a possible loop in the substitutions, which would otherwise put TEX in an infinite loop if just an exhaustive expansion was used.

To detect a loop, the *Tortoise and Hare* algorithm is used. The name of the algorithm is an analogy to Aesop's fable, in which the Hare outruns a Tortoise. The two pointers here are the csnames which contains each file replacement, both of which start at the position zero, which is the file requested. In the inner part of the macro below, `\__filehook_file_subst_loop:cc` is called with `\@file-subst@`⟨*file*⟩ and `\@file-subst@\@file-subst@`⟨*file*⟩; that is, the substitution of ⟨*file*⟩ and the substution of that substution: the Tortoise walks one step while the Hare walks two.

Within `\__filehook_file_subst_loop:NN` the two substitutions are compared, and if they lead to the same file it means that there is a loop in the substitutions. If there's no loop, `\__filehook_file_subst_tortoise_hare:nn` is called again with the Tortoise at position 1 and the hare at 2. Again, the substitutions are checked ahead of the Hare pointer to check that it won't run too far; in case there is no loop in the declarations, eventually one of the `\cs_if_exist:cTF` below will go ⟨*false*⟩ and the algorithm will end; otherwise it will run until the Hare reaches the same spot as the tortoise and a loop is detected.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\__filehook_file_subst_tortoise_hare:nn}{Tortoise and Hare}%
\ExplSyntaxOn
\cs_new:Npn \__filehook_file_subst_tortoise_hare:nn #1 #2 #3
  {
    \cs_if_exist:cTF { @file-subst@ #2 }
      {
        \flag_if_raised:nF { __filehook_file_replaced }
          { \flag_raise:n { __filehook_file_replaced } }
        \cs_if_exist:cTF { @file-subst@ \use:c { @file-subst@ #2 } }
          {
            \__filehook_file_subst_loop:cc
              { @file-subst@ #1 }
              { @file-subst@ \use:c { @file-subst@ #2 } }
          }
          { \use:c { @file-subst@ #2 } }
      }
      { #3 }
  }
```

This is just an auxiliary to check if a loop was found, and continue the algorithm otherwise. If a loop is found, the `.tex` file is used as fallback and `\__filehook_file_subst_cycle_error:cN` is called to report the error.

```
\cs_new:Npn \__filehook_file_subst_loop:NN #1 #2
  {
    \token_if_eq_meaning:NNTF #1 #2
      {
        .tex
        \__filehook_file_subst_cycle_error:cN { @file-subst@ #1 } #1
      }
      { \__filehook_file_subst_tortoise_hare:nn {#1} {#2} {#2} }
  }
\cs_generate_variant:Nn \__filehook_file_subst_loop:NN { cc }
```

__filehook_file_subst_cycle_error:NN
__filehook_file_subst_cycle_error:cN

Showing this type of error expandably is tricky, as we have a very limited amount of characters to show and a potentially large list. As a work around, several errors are printed, each showing one step of the loop, until all the error messages combined show the loop.

```
\cs_new:Npn \__filehook_file_subst_cycle_error:NN #1 #2
  {
    \__kernel_msg_expandable_error:nnff { kernel } { file-cycle }
      {#1} { \use:c { @file-subst@ #1 } }
    \token_if_eq_meaning:NNF #1 #2
      { \__filehook_file_subst_cycle_error:cN { @file-subst@ #1 } #2 }
  }
\cs_generate_variant:Nn \__filehook_file_subst_cycle_error:NN { c }
```

And the error message:

```
\__kernel_msg_new:nnn { kernel } { file-cycle }
  { File~loop!~#1~replaced~by~#2... }
```

(*End definition for* `\__filehook_resolve_file_subst:w` *and others.*)

```
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
⟨@@=⟩
```

2.8 Preventing a package from loading

We support the use case of preventing a package from loading but not any other type of files (e.g., classes).

\disable@package@load
\reenable@package@load
\@disable@packageload@do

`\disable@package@load` defines `\@pkg-disable@`⟨*package*⟩ to expand to some code `#2` instead of loading the package.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩          {\disable@package@load}{Disable packages}%
\def\disable@package@load#1#2{%
  \global\@namedef{@pkg-disable@#1.\@pkgextension}{#2}}
```

```
\def\@disable@packageload@do#1#2{%
  \@ifundefined{@pkg-disable@#1}{#2}%
     {\@nameuse{@pkg-disable@#1}}}
```

`\reenable@package@load` undefines `\@pkg-disable@`⟨*package*⟩ to reallow loading a package.

```
\def\reenable@package@load#1{%
  \global\expandafter\let
  \csname @pkg-disable@#1.\@pkgextension \endcsname \@undefined}
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩          {\disable@package@load}{Disable packages}%
⟨latexrelease⟩
⟨latexrelease⟩\let\disable@package@load   \@undefined
```

```
⟨latexrelease⟩\let\@disable@packageload@do\@undefined
⟨latexrelease⟩\let\reenable@package@load  \@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\disable@package@load`, `\reenable@package@load`, *and* `\@disable@packageload@do`. *These functions are documented on page* 683.)

2.9 High-level interfaces for LaTeX

None so far and the general feeling for now is that the hooks are enough. Packages like filehook, etc., may use them to set up their interfaces (samples are given below) but for the now the kernel will not provide any.

2.10 Internal commands needed elsewhere

Here we set up a few horrible (but consistent) LaTeX 2_ε names to allow for internal commands to be used outside this module (and in parts that still use LaTeX 2_ε syntax. We have to unset the `@@` since we want double "at" sign in place of double underscores.

```
⟨@@=⟩
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩    {\@expl@@@filehook@if@no@extension@@nTF}{2e tmp interfaces}%
\ExplSyntaxOn
\cs_new_eq:NN \@expl@@@filehook@if@no@extension@@nTF
              \__filehook_if_no_extension:nTF
\cs_new_eq:NN \@expl@@@filehook@set@curr@file@@nNN
              \__filehook_set_curr_file:nNN
\cs_new_eq:NN \@expl@@@filehook@resolve@file@subst@@w
              \__filehook_resolve_file_subst:w
\cs_new_eq:NN \@expl@@@filehook@normalize@file@name@@w
              \__filehook_normalize_file_name:w
\cs_new_eq:NN \@expl@@@filehook@if@file@replaced@@TF
              \__filehook_if_file_replaced:TF
\cs_new_eq:NN \@expl@@@filehook@clear@replacement@flag@@
              \__filehook_clear_replacement_flag:
\cs_new_eq:NN \@expl@@@filehook@drop@extension@@N
              \__filehook_drop_extension:N
\cs_new_eq:NN \@expl@@@filehook@file@push@@
               \__filehook_file_push:
\cs_new_eq:NN \@expl@@@filehook@file@pop@@
              \__filehook_file_pop:
\cs_new_eq:NN \@expl@@@filehook@file@pop@assign@@nnnn
              \__filehook_file_pop_assign:nnnn
\ExplSyntaxOff
```

```
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

This ends the kernel code in this file.

```
⟨/2ekernel⟩
```

3 A sample package for structuring the log output

```
⟨*structuredlog⟩
⟨@@=filehook⟩
```

```
\ProvidesExplPackage
    {structuredlog}{\ltfilehookdate}{\ltfilehookversion}
    {Structuring the TeX transcript file}
```

\g__filehook_nesting_level_int

Stores the current package nesting level.

```
\int_new:N \g__filehook_nesting_level_int
```

Initialise the counter with the number of files in the `\@currnamestack` (the number of items divided by 3) minus one, because this package is skipped when printing to the log.

```
\int_gset:Nn \g__filehook_nesting_level_int
  { ( \tl_count:N \@currnamestack ) / 3 - 1 }
```

(*End definition for* `\g__filehook_nesting_level_int`.)

__filehook_log_file_record:n

This macro is responsible for increasing and decresing the file nesting level, as well as printing to the log. The argument is either `STOPTART` or `STOP` and the action it takes on the nesting integer depends on that.

```
\cs_new_protected:Npn \__filehook_log_file_record:n #1
  {
    \str_if_eq:nnT {#1} {START} { \int_gincr:N \g__filehook_nesting_level_int }
    \iow_term:x
      {
        \prg_replicate:nn { \g__filehook_nesting_level_int } { = } ~
        ( LEVEL ~ \int_use:N \g__filehook_nesting_level_int \c_space_tl #1 ) ~
        \CurrentFileUsed
```

If there was a file replacement, show that as well:

```
        \str_if_eq:NNF \CurrentFileUsed \CurrentFile
          { ~ ( \CurrentFile \c_space_tl requested ) }
        \iow_newline:
      }
    \str_if_eq:nnT {#1} {STOP} { \int_gdecr:N \g__filehook_nesting_level_int }
  }
```

Now just hook the macro above in the generic `file/before`...

```
\AddToHook{file/before}{ \__filehook_log_file_record:n { START } }
```

...and `file/after` hooks. We don't want to install the `file/after` hook immediately, because that would mean it is the first time executed when the package finishes. We therefore put the declaration inside `\AddToHookNext` so that it gets only installed when we have left this package.

```
\AddToHookNext{file/after}
  { \AddToHook{file/after}{ \__filehook_log_file_record:n { STOP } } }
```

(*End definition for* `\__filehook_log_file_record:n`.)

```
⟨@@=⟩
⟨/structuredlog⟩
```

4 Package emulations

4.1 Package **atveryend** emulation

With the new hook management and the hooks in `\enddocument` all of atveryend is taken care of. We can make an emulation only here after the substitution functionality is available:

```
⟨*2ekernel⟩
\declare@file@substitution{atveryend.sty}{atveryend-ltx.sty}
⟨/2ekernel⟩
```

Here is the package file we point to:

```
⟨*atveryend-ltx⟩
\ProvidesPackage{atveryend-ltx}
   [2020/08/19 v1.0a
     Emulation of the original atvery package^^Jwith kernel methods]
```

Here are new definitions for its interfaces now pointing to the hooks in `\enddocument`

```
\newcommand\AfterLastShipout  {\AddToHook{enddocument/afterlastpage}}
\newcommand\AtVeryEndDocument {\AddToHook{enddocument/afteraux}}
```

Next one is a bit of a fake, but the result should normally be as expected. If not, one needs to add a rule to sort the code chunks in `enddocument/info`.

```
\newcommand\AtEndAfterFileList{\AddToHook{enddocument/info}}
\newcommand\AtVeryVeryEnd     {\AddToHook{enddocument/end}}
```

`\BeforeClearDocument` This one is the only one we don't implement or rather don't have a dedicated hook in the code.

```
\ExplSyntaxOn
\newcommand\BeforeClearDocument[1]
  { \AtEndDocument{#1}
    \atveryend@DEPRECATED{BeforeClearDocument \tl_to_str:n{#1}}
  }
\cs_new:Npn\atveryend@DEPRECATED #1
   {\iow_term:x{======~DEPRECATED~USAGE~#1~==========}}
\ExplSyntaxOff
```

(*End definition for* `\BeforeClearDocument`. *This function is documented on page* **??**.)

```
⟨/atveryend-ltx⟩
```

File R
ltshipout.dtx

Contents

1 Introduction

The code provides an interface to the `\shipout` primitive of TEX which is called when a finished pages is finally "shipped out" to the target output file, e.g., the `.dvi` or `.pdf` file. A good portion of the code is based on ideas by Heiko Oberdiek implemented in his packages atbegshi and atenddvi even though the interfaces are somewhat different.[21]

1.1 Overloading the `\shipout` primitive

`\shipout`

With this implementation TEX's shipout primitive is no longer available for direct use. Instead `\shipout` is running some (complicated) code that picks up the box to be shipped out regardless of how that is done, i.e., as a constructed `\vbox` or `\hbox` or as a box register.

It then stores it in a named box register. This box can then be manipulated through a set of hooks after which it is shipped out for real.

`\ShipoutBox`
`\l_shipout_box`

This box register is called `\ShipoutBox` (alternatively available via the L3 name `\l_shipout_box`).

`\l_shipout_box_ht_dim`
`\l_shipout_box_dp_dim`
`\l_shipout_box_wd_dim`
`\l_shipout_box_ht_plus_dp_dim`

The shipout box dimensions are available in the L3 registers `\l_shipout_box_ht_dim`, etc. (there are no LATEX 2_ε names).[22] These variables can be used inside the hook code for `shipout/before`, `shipout/foreground` and `shipout/background` if needed.

[21] Heiko's interfaces are emulated by the kernel code, if a document requests his packages, so older documents will continue to work.

[22] Might need changing, but HO's version as strings is not really helpful I think).

1.2 Provided hooks

shipout/before
shipout/foreground
shipout/background
shipout/firstpage
shipout/lastpage

The code offers a number of hooks into which packages (or the user) can add code to support different use cases. These are:

shipout/before This hook is executed after the finished page has been stored in `\ShipoutBox` / `\l_shipout_box`). It can be used to alter that box content or to discard it completely (see `\DiscardShipoutBox` below).

shipout/background This hook adds a picture environment into the background of the page with the `(0,0)` coordinate in the top-left corner using a `\unitlength` of `1pt`.

It should therefore only receive `\put` commands or other commands suitable in a `picture` environment and the vertical coordinate values would normally be negative.

Technically this is implemented by adding a zero-sized `\hbox` as the very first item into the `\ShipoutBox` containing that `picture` environment. Thus the rest of the box content will overprint what ever is typeset by that hook.

shipout/foreground This hook adds a picture environment into the foreground of the page with the `(0,0)` coordinate in the top-left corner using a `\unitlength` of `1pt`.

Technically this is implemented by adding a zero-sized `\hbox` as the very last item into the `\ShipoutBox` and raising it up so that it still has its `(0,0)` point in the top-left corner. But being placed after the main box content it will be typeset later and thus overprints it (i.e., is in the foreground).

shipout/firstpage The material from this hook is executed only once at the very beginning of the first output page that is shipped out (i.e., not discarded at the last minute). It should only contain `\special` or similar commands needed to direct post processors handling the `.dvi` or `.pdf` output.[23]

shipout/lastpage The corresponding hook to add `\special`s at the very end of the output file. It is only executed on the very last page — or rather on the page that LaTeX believes is the last one.

It may not be possible for LaTeX to correctly determine which page is the last one without several reruns. If this happens and the hook is non-empty then LaTeX will add an extra page to place the material and also request a rerun to get the correct placement sorted out.

As mentioned above the hook `shipout/before` is executed first and can manipulate the prepared shipout box stored in `\ShipoutBox` or set things up for use in `\write` during the actual shipout. The other hooks are added inside hboxes to the box being shipped out in the following order:

[23] In LaTeX 2_ε that was already existing, but implemented using a box register with the name `\@begindvibox`.

`shipout/firstpage`	only on the first page
`shipout/background`	
⟨*boxed content of* `\ShipoutBox`⟩	
`shipout/foreground`	
`shipout/lastpage`	only on the last page

If any of the hooks has no code then that particular no box is added at that point.

In a document that doesn't produce pages, e.g., only makes `\typeout`s, none of the hooks are executed (as there is no `\shipout`) not even the `shipout/lastpage` hook.

`\AtBeginDvi`
`\AtEndDvi`

`\AtBeginDvi` is the existing LaTeX 2_ε interface to fill the `shipout/firstpage` hook. This is not really a good name as it is not just supporting `.dvi` but also `.pdf` output or `.dvx`.

`\AtEndDvi` is the counterpart that was not available in the kernel but only through the package atenddvi. It fills the `shipout/lastpage` hook.

Todo: better names? Any suggestions?

1.3 Special commands for use inside the hooks

`\DiscardShipoutBox`
`\shipout_discard_box:`

`\AddToHookNext {shipout/before} {...\DiscardShipoutBox...}`

The `\DiscardShipoutBox` declaration (L3 name `\shipout_discard_box:`) requests that on the next shipout the page box is thrown away instead of being shipped to the `.dvi` or `.pdf` file.

Typical applications wouldn't do this unconditionally, but have some processing logic that decides to use or not to use the page.

Note that if this declaration is used directly in the document it may depend on the placement to which page it applies, given that LaTeX output routine is called in an asynchronous manner!

Todo: Once we have a new mark mechanism available we can improve on that and make sure that the declaration applies to the page that contains it.

In the atbegshi package there are a number of additional commands for use inside the `shipout/before` hook. They should normally not be needed any more as one can instead simply add code to the hooks `shipout/before`, `shipout/background` or `shipout/foreground`.[24] If atbegshi gets loaded then those commands become available as public functions with their original names as given below.

[24] If that assumption turns out to be wrong it would be trivial to change them to public functions (right now they are private).

1.4 Information counters

\ReadonlyShipoutCounter
\g_shipout_readonly_int

```
\ifnum\ReadOnlyShipoutCounter=...
\int_use:N \g_shipout_readonly_int % expl3 usage
```

This integer holds the number of pages shipped out up to now (including the one to be shipped out when inside the output routine). More precisely, it is incremented only after it is clear that a page will be shipped out, i.e., after the `shipout/before` hook (because that might discard the page)!

Just like with the `page` counter its value is only accurate within the output routine. In the body of the document it may be off by one as the output routine is called asynchronously!

Also important: it *must not* be set, only read. There are no provisions to prevent that but if you do, chaos will be the result. To emphasize this fact it is not provided as a LaTeX counter but as a TeX counter (i.e., a command), so `\Alph{⟨\ReadonlyShipoutCounter⟩}` etc, would not work.

totalpages
\g_shipout_totalpages_int

```
\arabic{totalpages}
\int_use:N \g_shipout_totalpage_int % expl3 usage
```

In contrast to `\ReadonlyShipoutCounter`, the `totalpages` counter is a LaTeX counter and incremented for each shipout attempt including those pages that are discarded for one or the other reason. Again `shipout/before` sees the counter before it is incremented).

Furthermore, while it is incremented for each page, its value is never used by LaTeX. It can therefore be freely reset or changed by user code, for example, to additionally count a number of pages that are not build by LaTeX but are added in a later part of the process, e.g., cover pages or picture pages made externally.

Important: as this is a page-related counter its value is only reliable inside the output routine!

\PreviousTotalPages

```
\thetotalpages/\PreviousTotalPages
```

Command that expands to the number of total pages from the previous run. If there was no previous run or if usedin the preamble it expands to `0`. Note that this is a command to a counter, so in order to display the number in, say, Roman numerals you have to assign its value to a counter and then use `\Roman` on that counter.

1.5 Debugging shipout code

\DebugShipoutsOn
\DebugShipoutsOff
\shipout_debug_on:
\shipout_debug_off:

```
\DebugShipoutsOn
```

Turn the debugging of shipout code on or off. This displays changes made to the shipout data structures.

Todo: This needs some rationalizing and may not stay this way.

2 Emulating commands from other packages

The packages in this section are no longer necessary but as they are used in other packages they are emulated when they are loaded via `\usepackage` or `\RequirePackage`.

2.1 Emulating atbegshi

`\AtBeginShipoutUpperLeft`
`\AtBeginShipoutUpperLeftForeground`

`\AddToHook {shipout/before}`
`{...\AtBeginShipoutUpperLeft{⟨code⟩}...}`

This adds a `picture` environment into the background of the shipout box expecting ⟨*code*⟩ to contain `picture` commands. The same effect can be obtained by simply using kernel features as follows:

`\AddToHook{shipout/background}{⟨code⟩}`

There is one technical difference: if `\AtBeginShipoutUpperLeft` is used several times each invocation is put into its own box inside the shipout box whereas all ⟨*code*⟩ going into `shipout/background` ends up all in the same box in the order it is added or sorted based on the rules for the hook chunks.

`\AtBeginShipoutUpperLeftForeground` is similar with the difference that the `picture` environment is placed in the foreground. To model it with the kernel functions use the hook `shipout/foreground` instead.

`\AtBeginShipoutAddToBox`
`\AtBeginShipoutAddToBoxForeground`

`\AddToHook {shipout/before} {...\AtBeginShipoutAddToBox{⟨code⟩}...}`

These work like `\AtBeginShipoutUpperLeft` and `\AtBeginShipoutUpperLeftForeground` with the difference that ⟨*code*⟩ is directly placed into an `\hbox` inside the shipout box and not surrounded by a `picture` environment.

To emulate them using `shipout/background` or `shipout/foreground` you may have to wrap ⟨*code*⟩ into a `\put` statement but if the code is not doing any typesetting just adding it to the hook should be sufficient.

`\AtBeginShipoutBox`

This is the name of the shipout box as atbegshi knows it.

`\AtBeginShipoutInit`

By default atbegshi delayed its action until `\begin{document}`. This command was forcing it in an earlier place. With the new concept it does nothing.

`\AtBeginShipout`
`\AtBeginShipoutNext`

`\AtBeginShipout{⟨code⟩}` ≡ `\AddToHook{shipout/before}{⟨code⟩}`
`\AtBeginShipoutNext{⟨code⟩}` ≡ `\AddToHookNext{shipout/before}{⟨code⟩}`

This is equivalent to filling the `shipout/before` hook by either using `\AddToHook` or `\AddToHookNext`, respectively.

`\AtBeginShipoutFirst`
`\AtBeginShipoutDiscard`

The atbegshi names for `\AtBeginDvi` and `\DiscardShipoutBox`.

2.2 Emulating everyshi

`\EveryShipout`

`\EveryShipout{⟨code⟩}` ≡ `\AddToHook{shipout/before}{⟨code⟩}`

`\AtNextShipout`

`\AtNextShipout{⟨code⟩}` ≡ `\AddToHookNext{shipout/before}{⟨code⟩}`

2.3 Emulating atenddvi

The atenddvi package implemented only a single command: `\AtEndDvi` and that is now available out of the box.

2.4 Emulating everypage

This page takes over the original `\@begindvi` hook and replaces it. It should be all covered by the hooks offered here (details need checking) and thus could simply use the provided hooks rather than defining its own.

3 The Implementation

```
⟨@@=shipout⟩
```

At the moment the whole module rolls back in one go, but if we make any modifications in later releases this will then need splitting.

```
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\shipout}{Hook mangement (shipout)}%
\ExplSyntaxOn
```

3.1 Debugging

`\g__shipout_debug_bool` Holds the current debugging state.

```
\bool_new:N \g__shipout_debug_bool
```

(*End definition for* `\g__shipout_debug_bool`.)

`\shipout_debug_on:` `\shipout_debug_off:` `\__shipout_debug:n` `\__shipout_debug_gset:` Turns debugging on and off by redefining `\__shipout_debug:n`.

```
\cs_new_eq:NN \__shipout_debug:n  \use_none:n
\cs_new_protected:Npn \shipout_debug_on:
  {
    \bool_gset_true:N \g__shipout_debug_bool
    \__shipout_debug_gset:
  }
\cs_new_protected:Npn \shipout_debug_off:
  {
    \bool_gset_false:N \g__shipout_debug_bool
    \__shipout_debug_gset:
  }
\cs_new_protected:Npn \__shipout_debug_gset:
  {
    \cs_gset_protected:Npx \__shipout_debug:n ##1
      { \bool_if:NT \g__shipout_debug_bool {##1} }
  }
```

(*End definition for* `\shipout_debug_on:` *and others. These functions are documented on page* 703.)

`\ShipoutBox` `\l_shipout_box` The box filled with the page to be shipped out (both L3 and LaTeX 2_ε name).

```
\box_new:N  \l_shipout_box
\cs_set_eq:NN \ShipoutBox \l_shipout_box
```

(*End definition for* `\ShipoutBox` *and* `\l_shipout_box`. *These functions are documented on page* 700.)

__shipout_execute: This is going to the be the code run by \shipout. The code follows closely the ideas from atbegshi, so not documenting that here for now.

```
\cs_set_protected:Npn \__shipout_execute: {
  \tl_set:Nx \l__shipout_group_level_tl
     { \int_value:w \tex_currentgrouplevel:D }
  \tex_afterassignment:D \__shipout_execute_test_level:
  \tex_setbox:D \l_shipout_box
}
```

(*End definition for* __shipout_execute:.)

\shipout Overloading the \shipout primitive:

```
\cs_gset_eq:NN \shipout \__shipout_execute:
```

(*End definition for* \shipout. *This function is documented on page* 700.)

\l__shipout_group_level_tl Helper token list to record the group level at which __shipout_execute: is encountered.

```
\tl_new:N \l__shipout_group_level_tl
```

(*End definition for* \l__shipout_group_level_tl.)

__shipout_execute_test_level: If the group level has changed then we are still constructing \l_shipout_box and to continue we need to wait until the current group has finished, hence the \tex_aftergroup:D.

```
\cs_new:Npn \__shipout_execute_test_level: {
  \int_compare:nNnT
     \l__shipout_group_level_tl < \tex_currentgrouplevel:D
     \tex_aftergroup:D
  \__shipout_execute_cont:
}
```

(*End definition for* __shipout_execute_test_level:.)

__shipout_execute_cont: When we have reached this point the shipout box has been processed and is available in \l_shipout_box and ready for real ship out (perhaps)..

First we quickly check if it is void (can't happen in the standard LaTeX output routine but \shipout might be called from a package that has some special processing logic). If it is void we aren't shipping anything out and processing ends.[25]

```
\cs_new:Npn \__shipout_execute_cont: {
  \box_if_empty:NTF \l_shipout_box
    { \PackageWarning{ltshipout}{Ignoring~ void~ shipout~ box} }
    {
```

Otherwise we assume that we will ship something and prepare for final adjustments (in particular setting the state of \protect while we are running the hook code). We also save the current \protect state to restore it later.

```
      \bool_gset_false:N \g__shipout_discard_bool
      \cs_set_eq:NN \__shipout_saved_protect: \protect
      \set@typeset@protect
```

We also store the current shipout box dimension in registers, so that they can be used in the hook code.[26]

```
      \__shipout_get_box_size:N \l_shipout_box
```

[25] In that case we don't reset the deadcyles, that would be up to the OR processing logic to do.

[26] This is not really necessary as the code could access them via \box_ht:N, etc., but it is perhaps convenient.

Then we execute the `shipout/before` hook.

```
      \hook_use:n {shipout/before}
```

In `\g_shipout_totalpages_int` we count all shipout attempts so we increment that counter already here (the other one is incremented later when we know for sure that we do a `\shipout`.

We increment it after running the above hook so that the values for `\g_shipout_totalpages_int` and `\` are in sync while the hook is executed (in the case that totalpages isn't manually altered or through discarding pages that is).

```
      \int_gincr:N \g_shipout_totalpages_int
```

The above hook might contain code that requests the page to be discarded so we now test for it.

```
      \bool_if:NTF \g__shipout_discard_bool
        { \PackageInfo{ltshipout}{Completed~ page~ discarded}
          \bool_gset_false:N \g__shipout_discard_bool
```

As we are discarding the page box and not shipping anything out, we need to do some house cleaning and reset TEX's deadcycles so that it doesn't complain about too many calls to the OR without any shipout.

```
          \tex_deadcycles:D \c_zero_int
```

Todo: In atbegshi *the box was dropped but is that actually needed? Or the resetting of* `\protect` *to its kernel value?*

```
%           \group_begin:
%             \box_set_eq_drop:NN \l_shipout_box \l_shipout_box
%           \group_end:
%           \cs_set_eq:NN \protect \exp_not:N
        }
```

Even if there was no explicit request to discard the box it is possible that the code for the hook `shipout/before` has voided the box (by mistake or deliberately). We therefore test once more but this time make it a warning, because the best practice way is to use the request mechanism.

```
        { \box_if_empty:NTF \l_shipout_box
            { \PackageWarning{ltshipout}{
                Shipout~ box~ was~ voided~ by~ hook,\MessageBreak
                ignoring~ shipout~ box  }
            }
```

Finally, if the box is still non-empty we are nearly ready to ship it out. First we increment the total page counter so that we can later test if we have reached the final page according to our available information.[27]

```
            {
              \int_gincr:N \g_shipout_readonly_int
              \__shipout_debug:n {
                \typeout{Absolute~ page~ =~ \int_use:N \g_shipout_readonly_int
                         \space (target:~ \@abspage@last)}
              }
```

[27] Doing that earlier would be wrong because we might end up with the last page counted but discard and then we have no place to add the final objects into the output file.

Then we store the box sizes again (as they may have changed) and then look at the hooks `shipout/foreground` and `shipout/background`. If either or both are non-empty we add a `picture` environment to the box (in the foreground and or in the background) and execute the hook code inside that environment.

```
            \__shipout_get_box_size:N \l_shipout_box
            \hook_if_empty:nF {shipout/foreground}
                 { \__shipout_add_foreground_picture:n
                   { \hook_use:n {shipout/foreground} } }
            \hook_if_empty:nF {shipout/background}
                 { \__shipout_add_background_picture:n
                   { \@kernel@before@shipout@background
                     \hook_use:n {shipout/background} } }
```

We then run `\__shipout_execute_firstpage_hook:` that adds the content of the hook `shipout/firstpage` to the start of the first page (if non-empty). It is then redefined to do nothing on later pages.

```
            \__shipout_execute_firstpage_hook:
```

The we check if we have to add the `shipout/lastpage` hook because we have reached the last page. This test will be false for all but one (and hopefully the correct) page.

```
            \int_compare:nNnT \@abspage@last = \g_shipout_readonly_int
              { \hook_if_empty:nF {shipout/lastpage}
                  { \__shipout_debug:n { \typeout{Executing~ lastpage~ hook~
                        on~ page~ \int_use:N \g_shipout_readonly_int }        }
                    \__shipout_add_foreground_box:n { \UseHook{shipout/lastpage}
                                                  \@kernel@after@shipout@lastpage }
                  }
                  \bool_gset_true:N \g__shipout_lastpage_handled_bool
              }
```

Finally we run the actual TEX primitive for shipout. As that will expand delayed `\write` statements inside the page in which protected commands should not expand we first change `\protect` to the appropriate definition for that case.

```
            \cs_set_eq:NN \protect \exp_not:N
            \tex_shipout:D \box_use:N \l_shipout_box
          }
      }
```

Restore the value of `\protect` in case `\shipout` is called outside of the output routine (where it is automatically restored because of the implicit group).

```
      \cs_set_eq:NN \protect \__shipout_saved_protect:
    }
}
```

(*End definition for* `\__shipout_execute_cont:`.)

`\__shipout_saved_protect:` Remeber the current `\protect` state.

```
\cs_new_eq:NN  \__shipout_saved_protect: \protect
```

(*End definition for* `\__shipout_saved_protect:`.)

`shipout/before`
`shipout/foreground`
`shipout/background`
`shipout/firstpage`
`shipout/lastpage`

Declaring all hooks for the shipout code.

```
\hook_new:n{shipout/before}
\hook_new:n{shipout/foreground}
\hook_new:n{shipout/background}
```

```
\hook_new:n{shipout/firstpage}
\hook_new:n{shipout/lastpage}
```

(*End definition for* `shipout/before` *and others. These functions are documented on page* 701.)

\@kernel@after@shipout@lastpage
\@kernel@before@shipout@background

And here are the internal kernel hooks going before or after the public ones where needed.

```
\let\@kernel@after@shipout@lastpage\@empty
\let\@kernel@before@shipout@background\@empty
```

(*End definition for* `\@kernel@after@shipout@lastpage` *and* `\@kernel@before@shipout@background`*. These functions are documented on page* ??.)

__shipout_execute_firstpage_hook:

This command adds any specials into a box and adds that to the very beginning of the first box shipped out. After that we redefine it to do nothing on later pages.

```
\cs_new:Npn \__shipout_execute_firstpage_hook: {
```

Adding something to the beginning means adding it to the background as that layer is done first in the output. Of course that is only needed if the hook actually contains anything.

```
  \hook_if_empty:nF {shipout/firstpage}
       { \__shipout_add_background_box:n { \UseHook{shipout/firstpage} } }
```

Once we are here we change the definition to do nothing next time and we also change the command used to implement `\AtBeginDvi` to become a warning and not add further material to a hook that is never used again.

```
  \cs_gset_eq:NN \__shipout_execute_firstpage_hook: \prg_do_nothing:
  \cs_gset:Npn \__shipout_add_firstpage_material:Nn ##1 ##2 {
    \@latex@warning{
        First~ page~ is~ already~ shipped~ out,~ ignoring\MessageBreak
        \string##1 }
  }
}
```

(*End definition for* `\__shipout_execute_firstpage_hook:`.)

\g__shipout_lastpage_handled_bool

A boolean to signal if we have already handled the `shipout/lastpage` hook.

```
\bool_new:N \g__shipout_lastpage_handled_bool
```

(*End definition for* `\g__shipout_lastpage_handled_bool`.)

__shipout_add_firstpage_material:Nn

This command adds material to the `shipout/firstpage` hook. It is used in `\AtBeginDvi`, etc. The first argument is the command through which is it called. Initially this is ignored but once we are passed the first page it can be used to generate a warning message mentioning the right user command.

```
\cs_new:Npn \__shipout_add_firstpage_material:Nn #1#2 {
   \AddToHook{shipout/firstpage}{#2}
}
```

(*End definition for* `\__shipout_add_firstpage_material:Nn`.)

__shipout_get_box_size:N

Store the box dimensions in dimen registers.

> *Todo: This could/should perhaps be generalized to set height depth and width given an arbitrary box.*

```
\cs_new:Npn \__shipout_get_box_size:N #1 {
  \dim_set:Nn \l_shipout_box_ht_dim { \box_ht:N #1 }
  \dim_set:Nn \l_shipout_box_dp_dim { \box_dp:N #1 }
  \dim_set:Nn \l_shipout_box_wd_dim { \box_wd:N #1 }
  \dim_set:Nn \l_shipout_box_ht_plus_dp_dim { \l_shipout_box_ht_dim +
                                              \l_shipout_box_dp_dim }
}
```

(*End definition for* `\__shipout_get_box_size:N`.)

`\l_shipout_box_ht_dim`
`\l_shipout_box_dp_dim`
`\l_shipout_box_wd_dim`
`\l_shipout_box_ht_plus_dp_dim`

And here are the variables set by `\__shipout_get_box_size:N`.

```
\dim_new:N \l_shipout_box_ht_dim
\dim_new:N \l_shipout_box_dp_dim
\dim_new:N \l_shipout_box_wd_dim
\dim_new:N \l_shipout_box_ht_plus_dp_dim
```

(*End definition for* `\l_shipout_box_ht_dim` *and others. These functions are documented on page* 700.)

`\g__shipout_discard_bool`

Indicate whether or not the current page box should be discarded

```
\bool_new:N \g__shipout_discard_bool
```

(*End definition for* `\g__shipout_discard_bool`.)

`\l__shipout_tmp_box`
`\l__shipout_saved_badness_tl`

We need a box for the background and foreground material and a token register to remember badness settings as we disable them during te buildup below.

```
\box_new:N \l__shipout_tmp_box
\tl_new:N  \l__shipout_saved_badness_tl
```

(*End definition for* `\l__shipout_tmp_box` *and* `\l__shipout_saved_badness_tl`.)

`\__shipout_add_background_box:n`

In standard LaTeX the shipout box is always a `\vbox` but here we are allow for other usage as well, in case some package has its own output routine.

```
\cs_new:Npn \__shipout_add_background_box:n #1
{ \__shipout_get_box_size:N \l_shipout_box
```

But we start testing for a vertical box as that should be the normal case.

```
  \box_if_vertical:NTF \l_shipout_box
      {
```

Save current values of `\vfuzz` and `\vbadness` then change them to allow box manipulations without warnings.

```
        \tl_set:Nx \l__shipout_saved_badness_tl
           { \vfuzz=\the\vfuzz\relax
             \vbadness=\the\vbadness\relax }
        \vfuzz=\c_max_dim
        \vbadness=\c_max_int
```

Then we reconstruct `\l_shipout_box` ...

```
        \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
             {
```

... the material in `#1` is placed into a horizontal box with zero dimensions.

```
                \hbox_set:Nn \l__shipout_tmp_box
                     { \l__shipout_saved_badness_tl #1 }
                \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
                \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
                \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
```

The we typeset that box followed by whatever was in `\l_shipout_box` before (unpacked).

```
                \skip_zero:N \baselineskip
                \skip_zero:N \lineskip
                \skip_zero:N \lineskiplimit
                \box_use:N \l__shipout_tmp_box
                \vbox_unpack:N \l_shipout_box
```

The `\kern` ensures that the box has no depth which is afterwards explicitly corrected.

```
                \kern \c_zero_dim
              }
        \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
        \box_set_dp:Nn \l_shipout_box \l_shipout_box_dp_dim
```

Todo: The whole boxing maneuver looks a bit like overkill to me, but for the moment I leave.

```
        \l__shipout_saved_badness_tl
      }
      {
```

A horizontal box is handled in a similar way. The last case would be a void box in which case we do nothing hence the missing `F` branch.

```
        \box_if_horizontal:NT \l_shipout_box
            {
              \tl_set:Nx \l__shipout_saved_badness_tl
                 { \hfuzz=\the\hfuzz\relax
                   \hbadness=\the\hbadness\relax }
              \hfuzz=\c_max_dim
              \hbadness=\c_max_int
              \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
                   {
                     \hbox_set:Nn \l__shipout_tmp_box
                          { \l__shipout_saved_badness_tl #1 }
                     \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
                     \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
                     \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
                     \box_move_up:nn
                         \l_shipout_box_ht_dim
                         { \box_use:N \l__shipout_tmp_box }
                     \hbox_unpack:N \l_shipout_box
                   }
              \l__shipout_saved_badness_tl
            }
      }
}
```

(*End definition for* `\__shipout_add_background_box:n`.)

`\__shipout_add_foreground_box:n` Foreground boxes are done in the same way, only the order and placement of boxes has to be done differently.

```
\cs_new:Npn \__shipout_add_foreground_box:n #1
{
  \box_if_vertical:NTF \l_shipout_box
    {
      \tl_set:Nx \l__shipout_saved_badness_tl
```

```
        { \vfuzz=\the\vfuzz\relax
          \vbadness=\the\vbadness\relax }
      \vfuzz=\c_max_dim
      \vbadness=\c_max_int
      \vbox_set_to_ht:Nnn \l_shipout_box \l_shipout_box_ht_plus_dp_dim
           {
             \hbox_set:Nn \l__shipout_tmp_box
                  { \l__shipout_saved_badness_tl #1 }
             \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
             \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
             \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
             \skip_zero:N \baselineskip
             \skip_zero:N \lineskip
             \skip_zero:N \lineskiplimit
             \vbox_unpack:N \l_shipout_box
             \kern -\l_shipout_box_ht_plus_dp_dim
             \box_use:N \l__shipout_tmp_box
             \kern  \l_shipout_box_ht_plus_dp_dim
           }
      \l__shipout_saved_badness_tl
      \box_set_ht:Nn \l_shipout_box \l_shipout_box_ht_dim
      \box_set_dp:Nn \l_shipout_box \l_shipout_box_dp_dim
    }
    {
      \box_if_horizontal:NT \l_shipout_box
        {
          \tl_set:Nx \l__shipout_saved_badness_tl
            { \hfuzz=\the\hfuzz\relax
              \hbadness=\the\hbadness\relax }
          \hfuzz=\c_max_dim
          \hbadness=\c_max_int
          \hbox_set_to_wd:Nnn \l_shipout_box \l_shipout_box_wd_dim
               {
                 \hbox_unpack:N \l_shipout_box
                 \kern -\box_wd:N \l_shipout_box
                 \hbox_set:Nn \l__shipout_tmp_box
                     { \l__shipout_saved_badness_tl #1 }
                 \box_set_wd:Nn \l__shipout_tmp_box \c_zero_dim
                 \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
                 \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
                 \box_move_up:nn { \box_ht:N \l_shipout_box }
                               { \box_use:N \l__shipout_tmp_box }
                 \kern \box_wd:N \l_shipout_box
               }%
               \l__shipout_saved_badness_tl
        }
    }
}
```

(*End definition for* `\__shipout_add_foreground_box:n`.)

`\__shipout_init_page_origins:`
`\c__shipout_horigin_tl`
`\c__shipout_vorigin_tl`

Two constants holding the offset of the top-left with respect to the media box. Setting the constants this way is courtesy of Bruno.

We delay setting the constants to the last possible place as there might be updates in the preamble or even in the `begindocument` hook that affects their setup.

```
\cs_new:Npn \__shipout_init_page_origins: {
  \tl_const:Nx \c__shipout_horigin_tl
     {
       \cs_if_exist_use:NTF \pdfvariable { horigin }
          { \cs_if_exist_use:NF \pdfhorigin { 1in } }
     }
  \tl_const:Nx \c__shipout_vorigin_tl
     {
       \cs_if_exist_use:NTF \pdfvariable { vorigin }
          { \cs_if_exist_use:NF \pdfvorigin { 1in } }
     }
```

After the constants have been set there is no need to execute this command again, in fact it would raise an error, so we redefine it to do nothing.

```
  \cs_gset_eq:NN \__shipout_init_page_origins: \prg_do_nothing:
}
```

(*End definition for* `\__shipout_init_page_origins:`, `\c__shipout_horigin_tl`, *and* `\c__shipout_vorigin_tl`.)

`\__shipout_picture_overlay:n` Put the argument into a `picture` environment that doesn't take up any size and uses `1pt` for `\unitlength`.

> *Todo: Could perhaps be generalized as it might be useful elsewhere. For now it is not.*

```
\cs_new:Npn \__shipout_picture_overlay:n #1 {
```

The very first time this is executed we have to initializes (and freeze) the origins.

```
    \__shipout_init_page_origins:
    \kern -\c__shipout_horigin_tl \scan_stop:
    \vbox_to_zero:n {
      \kern -\c__shipout_vorigin_tl \scan_stop:
      \unitlength 1pt \scan_stop:
```

This mimics a simple zero-sized picture environment. The `\hss` is need in case there is horizontal material (without using `\put` with a positive width.

```
      \hbox_set_to_wd:Nnn \l__shipout_tmp_box \c_zero_dim
                          { \ignorespaces #1 \hss }
      \box_set_ht:Nn \l__shipout_tmp_box \c_zero_dim
      \box_set_dp:Nn \l__shipout_tmp_box \c_zero_dim
      \box_use:N \l__shipout_tmp_box
      \tex_vss:D
    }
}
```

(*End definition for* `\__shipout_picture_overlay:n`.)

`\__shipout_add_background_picture:n` Put a `picture` env in the background of the shipout box with its reference point in the top-left corner.

```
\cs_new:Npn \__shipout_add_background_picture:n #1 {
   \__shipout_add_background_box:n { \__shipout_picture_overlay:n {#1} }
}
```

(*End definition for* `\__shipout_add_background_picture:n`.)

`\__shipout_add_foreground_picture:n` Put a `picture` env in the foreground of the shipout box with its reference point in the top-left corner.

```
\cs_new:Npn \__shipout_add_foreground_picture:n #1 {
   \__shipout_add_foreground_box:n { \__shipout_picture_overlay:n {#1} }
}
```

(*End definition for* `\__shipout_add_foreground_picture:n`.)

`\shipout_discard:` Request that the next shipout box should be discarded. At the moment this is just setting a boolean, but we may want to augment this behavior that the position of the call is taken into account (in case LaTeX looks ahead and is not using the position for on the next page).

```
\cs_new_protected:Npn \shipout_discard: {
  \bool_gset_true:N \g__shipout_discard_bool
}
```

(*End definition for* `\shipout_discard:`. *This function is documented on page* **??**.)

3.2 Handling the end of job hook

At the moment this is partly solved by using the existing hooks. But rather than putting the code into these hooks it should be moved to the right place directly as we shouldn't prefill hooks with material unless it needs to interact with other code.

`\g_shipout_readonly_int` `\ReadonlyShipoutCounter` We count every shipout activity that makes a page (but not those that are discarded) in order to know how many pages got produced.

```
\int_new:N \g_shipout_readonly_int
```

For LaTeX 2_ε it is available as a command (i.e., a TeX counter only.

```
\cs_new_eq:NN \ReadonlyShipoutCounter  \g_shipout_readonly_int
```

(*End definition for* `\g_shipout_readonly_int` *and* `\ReadonlyShipoutCounter`. *These functions are documented on page* 703.)

`\g_shipout_totalpages_int` `\c@totalpages` We count every shipout attempt (even those that are discarded) in tis counter. It is not used in the code but may get used in user code.

```
\int_new:N \g_shipout_totalpages_int
```

For LaTeX 2_ε this is offered as a LaTeX counter so can be easily typeset inside the output routine to display things like "`\thepage/\thetotalpages`", etc.

```
\cs_new_eq:NN \c@totalpages \g_shipout_totalpages_int
\cs_new:Npn \thetotalpages { \arabic{totalpages} }
```

(*End definition for* `\g_shipout_totalpages_int` *and* `\c@totalpages`. *These functions are documented on page* 703.)

`\@abspage@last` In `\@abspage@last` record the number of pages from the last run. This is written to the `.aux` and this way made available to the next run. In case there is no `.aux` file or the statement is missing from it we initialize it with the largest possible number in TeX. We use this as the default because then we are inserting the `shipout/lastpage` on the last page (or after the last page but not on page 1 for a multipage document.

```
\xdef\@abspage@last{\number\maxdimen}
```

(*End definition for* `\@abspage@last`. *This function is documented on page* **??**.)

`\enddocument` Instead of using the hooks `enddocument` and `enddocument/afterlastpage` we add this code to private kernel hooks to be 100% when it is executed and to avoid cluttering the hooks with data that is always there.

Inside `\enddocument` there is a `\clearpage`. Just before that we execute this code here. There is a good change that we are on the last page. Therefore, if we don't know the value from the last run, we assume that the current page is the right one. So we set `\@abspage@last` and as a result the next shipout will run the `shipout/lastpage` code. Of course, if there are floats that still need a placement this guess will be wrong but then rerunning the document will give us the correct value next time around.

`\@kernel@after@enddocument`

```
\g@addto@macro \@kernel@after@enddocument {
  \int_compare:nNnT \@abspage@last = \maxdimen
    {
```

We use LaTeX 2_ε coding as `\@abspage@last` is not an L3 name.

```
      \xdef\@abspage@last{ \int_eval:n {\g_shipout_readonly_int + 1} }
    }
}
```

`\@kernel@after@enddocument@afterlastpage`

Once the `\clearpage` has done its work inside `\enddocument` we know for sure how many pages this document has, so we record that in the `.aux` file for the next run.

```
\g@addto@macro \@kernel@after@enddocument@afterlastpage {
```

There is one special case: If no output is produced then there is no point in a) recording the number as 0 will never match the page number of a real page and b) adding an extra page to ran the `shipout/lastpage` is pointless as well (as it would remain forever). So we test for this and run the code only if there have been pages.

```
  \int_compare:nNnF \g_shipout_readonly_int = 0
    {
```

This ends up in the `.aux` so we use LaTeX 2_ε names here.

Todo: This needs an interface for `\nofiles` *in expl3, doesn't at the moment!*

```
      \if@filesw
         \iow_now:Nx \@auxout {
           \gdef\string\@abspage@last {\int_use:N \g_shipout_readonly_int}}
      \fi
```

But we may have guessed wrongly earlier and we still have to run the `shipout/lastpage` even though there is no page to place it into. If that is the case we make a trivial extra page and put it there. This temporary page will then vanish again on the next run but helps to keep pdf viewers happy.

```
       \bool_if:NF \g__shipout_lastpage_handled_bool
          {
```

However, making this extra page in case the hook is actually empty would be forcing a rerun without any reason, so we check that condition and also check if `\@kernel@after@shipout@lastpage` contains any code. If both are empty we omit the page generation.

```
           \bool_lazy_and:nnF
             { \hook_if_empty_p:n {shipout/lastpage} }
```

```
            { \tl_if_empty_p:N \@kernel@after@shipout@lastpage }
            {
              \tex_shipout:D\vbox to\textheight
                {
                  \hbox:n { \UseHook{shipout/lastpage}
                            \@kernel@after@shipout@lastpage }
```

This extra page could be totally empty except for the hook content, but to help the user understanding why it is there we put some text into it.

```
                  \__shipout_excuse_extra_page:
                  \null
                }
```

At this point we also signal to LaTeX's endgame that a rerun is necessary so that an appropriate message can be shown on the terminal. We do this by simply defining a command used as a flag and tested `\enddocument`.

```
              \cs_gset_eq:NN \@extra@page@added \relax
            }
          }
      }
}
```

(*End definition for* `\enddocument`, `\@kernel@after@enddocument`, *and* `\@kernel@after@enddocument@afterlastpage`. *These functions are documented on page* **??**.)

`\__shipout_excuse_extra_page:` Say mea culpa ...

```
\cs_new:Npn \__shipout_excuse_extra_page: {
  \vfil
  \begin{center}
    \bfseries Temporary~ page!
  \end{center}
    \LaTeX{}~ was~ unable~ to~ guess~ the~ total~ number~ of~ pages~
    correctly.~ ~ As~ there~ was~ some~ unprocessed~ data~ that~
    should~ have~ been~ added~ to~ the~ final~ page~ this~ extra~
    page~ has~ been~ added~ to~ receive~ it.
    \par
    If~ you~ rerun~ the~ document~ (without~ altering~ it)~ this~
    surplus~ page~ will~ go~ away,~ because~ \LaTeX{}~ now~ knows~
    how~ many~ pages~ to~ expect~ for~ this~ document.
  \vfil
}
```

(*End definition for* `\__shipout_excuse_extra_page:`.)

`\PreviousTotalPages` `\@kernel@before@begindocument` In the preamble before the `aux` file was read `\PreviousTotalPages` is always zero.

```
\def\PreviousTotalPages{0}
```

In the `aux` file there should be an update for `\@abspage@last` recording the number of pages from the previous run. If not that macro holds the value of `\maxdimen`. So we test for it and update `\PreviousTotalPages` if there was a real value. This should happen just before the `begindocument` hook is executed so that the value can be used inside that hook.

```
\g@addto@macro\@kernel@before@begindocument
  {\ifnum\@abspage@last<\maxdimen
     \xdef\PreviousTotalPages{\@abspage@last}\fi}
```

(*End definition for* `\PreviousTotalPages` *and* `\@kernel@before@begindocument`*. These functions are documented on page* 703.)

4 Legacy LaTeX 2_ε interfaces

`\DiscardShipoutBox` Request that the next shipout box is to be discarded.

```
\cs_new_eq:NN \DiscardShipoutBox \shipout_discard:
```

(*End definition for* `\DiscardShipoutBox`*. This function is documented on page* 702.)

`\AtBeginDvi` If we roll forward from an earlier kernel `\AtBeginDvi` is defined so we better not use `\cs_new_protected:Npn` here.

```
\cs_set_protected:Npn \AtBeginDvi {\__shipout_add_firstpage_material:Nn \AtBeginDvi}
```

(*End definition for* `\AtBeginDvi`*. This function is documented on page* 702.)

`\DebugShipoutsOn`
`\DebugShipoutsOff`

```
\cs_new_eq:NN \DebugShipoutsOn  \shipout_debug_on:
\cs_new_eq:NN \DebugShipoutsOff \shipout_debug_off:
```

(*End definition for* `\DebugShipoutsOn` *and* `\DebugShipoutsOff`*. These functions are documented on page* 703.)

5 Internal commands needed elsewhere

These internal commands use double and triple `@` signs so we need to stop getting them translated to the module name.

```
⟨@@=⟩
```

`\@expl@@@shipout@add@firstpage@material@@Nn`
`\@expl@@@shipout@add@background@box@@n`
`\@expl@@@shipout@add@foreground@box@@n`
`\@expl@@@shipout@add@background@picture@@n`
`\@expl@@@shipout@add@foreground@picture@@n`

Some internals needed elsewhere.

```
\cs_set_eq:NN \@expl@@@shipout@add@firstpage@material@@Nn
              \__shipout_add_firstpage_material:Nn
\cs_set_eq:NN \@expl@@@shipout@add@background@box@@n
              \__shipout_add_background_box:n
\cs_set_eq:NN \@expl@@@shipout@add@foreground@box@@n
              \__shipout_add_foreground_box:n
\cs_set_eq:NN \@expl@@@shipout@add@background@picture@@n
              \__shipout_add_background_picture:n
\cs_set_eq:NN \@expl@@@shipout@add@foreground@picture@@n
              \__shipout_add_foreground_picture:n
```

(*End definition for* `\@expl@@@shipout@add@firstpage@material@@Nn` *and others. These functions are documented on page* ??.)

```
\ExplSyntaxOff
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

Rolling back here doesn't undefine the interface commands as they may be used in packages without rollback functionality. So we just make them do nothing which may or may not work depending on the code usage.

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                 {\shipout}{The hook management (shipout)}%
⟨latexrelease⟩
```

If we roll forward then `\tex_shipout:D` may not be defined in which case `\shipout` does have it original definition and so we must not `\let` it to something else which is `\relax`!

```
⟨latexrelease⟩\ifcsname tex_shipout:D\endcsname
⟨latexrelease⟩\expandafter\let\expandafter\shipout
⟨latexrelease⟩                \csname tex_shipout:D\endcsname
⟨latexrelease⟩\fi
⟨latexrelease⟩
⟨latexrelease⟩\let \ShipoutBox\@undefined
⟨latexrelease⟩\let \ReadonlyShipoutCounter \@undefined
⟨latexrelease⟩\let \c@totalpages \@undefined
⟨latexrelease⟩\let \thetotalpages \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let \DiscardShipoutBox \@undefined
⟨latexrelease⟩\let \DebugShipoutsOn \@undefined
⟨latexrelease⟩\let \DebugShipoutsOff \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\DeclareRobustCommand \AtBeginDvi [1]{%
⟨latexrelease⟩  \global \setbox \@begindvibox
⟨latexrelease⟩    \vbox{\unvbox \@begindvibox #1}%
⟨latexrelease⟩}
⟨latexrelease⟩
⟨latexrelease⟩\let \AtBeginShipout \@undefined
⟨latexrelease⟩\let \AtBeginShipoutNext \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let \AtBeginShipoutFirst \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let \ShipoutBoxHeight \@undefined
⟨latexrelease⟩\let \ShipoutBoxDepth \@undefined
⟨latexrelease⟩\let \ShipoutBoxWidth \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let \AtBeginShipoutDiscard \@undefined
⟨latexrelease⟩
⟨latexrelease⟩\let \AtBeginShipoutAddToBox \@undefined
⟨latexrelease⟩\let \AtBeginShipoutAddToBoxForeground \@undefined
⟨latexrelease⟩\let \AtBeginShipoutUpperLeft \@undefined
⟨latexrelease⟩\let \AtBeginShipoutUpperLeftForeground \@undefined
⟨latexrelease⟩
```

We do not undo a substitution when rolling back. As the file support gets undone the underlying data is no longer used (and sufficiently obscure that should not interfer with existing commands) and properly removing it would mean we need to make the `\unclare@...` and its support macros available in all earlier kernel releases which is pointless (and actually worse).

```
%\undeclare@file@substitution{everyshi.sty}
⟨latexrelease⟩
⟨latexrelease⟩\let  \AtEndDvi \@undefined
```

We do not reenable a disabled package load when rolling back. As the file support gets undone the underlying data is no longer checked (and sufficiently obscure that it should not interfer with existing commands) and properly removing it would mean we need to make the `\reenable@package@load` command available in all earlier kernel releases which is pointless (and actually worse).

```
%\reenable@package@load{atenddvi}
```

```
⟨latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

6 Package emulation for compatibility

6.1 Package atenddvi emulation

`\AtEndDvi` This package has only one public command to simulating it is easy and actually sensible to provide as part of the kernel.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2020/10/01}%
⟨latexrelease⟩                  {\AtEndDvi}{atenddvi emulation}%
\ExplSyntaxOn
\cs_new_protected:Npn \AtEndDvi {\AddToHook{shipout/lastpage}}
\ExplSyntaxOff
```

As the package is integrate we prevent loading (no need to roll that back):

```
\disable@package@load{atenddvi}
   {\PackageWarning{atenddvi}
     {Functionality of this package is already\MessageBreak
      provided by LaTeX.\MessageBreak\MessageBreak
      It is there no longer necessary to load it\MessageBreak
      and you can safely remove it.\MessageBreak
      Found on}}
```

```
⟨/2ekernel | latexrelease⟩
```

```
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\AtEndDvi}{atenddvi emulation}%
⟨latexrelease⟩\let \AtEndDvi \@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\AtEndDvi`*. This function is documented on page 702.*)

```
⟨/2ekernel⟩
```

6.2 Package atbegshi emulation

```
⟨*atbegshi-ltx⟩
\ProvidesPackage{atbegshi-ltx}
   [2020/08/17 v1.0a
     Emulation of the original atbegshi package^^Jwith kernel methods]
```

`\AtBeginShipoutBox`

```
\let \AtBeginShipoutBox \ShipoutBox
```

(*End definition for* `\AtBeginShipoutBox`. *This function is documented on page* 704.)

`\AtBeginShipoutInit` Compatibility only, we aren't delaying ...

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

(*End definition for* `\AtBeginShipoutInit`. *This function is documented on page* 704.)

`\AtBeginShipout` `\AtBeginShipoutNext` Filling hooks

```
\protected \def \AtBeginShipout     {\AddToHook{shipout/before}}
\protected \def \AtBeginShipoutNext {\AddToHookNext{shipout/before}}
```

(*End definition for* `\AtBeginShipout` *and* `\AtBeginShipoutNext`. *These functions are documented on page* 704.)

`\AtBeginShipoutFirst` Slightly more complex as we need to know the name of the command under which the `shipout/firstpage` hook is filled.

```
\protected \def \AtBeginShipoutFirst
   {\@expl@@@shipout@add@firstpage@material@@Nn \AtBeginShipoutFirst}
```

(*End definition for* `\AtBeginShipoutFirst`. *This function is documented on page* 704.)

`\AtBeginShipoutDiscard` Just a different name.

```
\let \AtBeginShipoutDiscard \DiscardShipoutBox
```

(*End definition for* `\AtBeginShipoutDiscard`. *This function is documented on page* 704.)

`\AtBeginShipoutAddToBox` `\AtBeginShipoutAddToBoxForeground` `\AtBeginShipoutUpperLeft` `\AtBeginShipoutUpperLeftForeground` We don't expose them.

```
\let \AtBeginShipoutAddToBox
              \@expl@@@shipout@add@background@box@@n
\let \AtBeginShipoutAddToBoxForeground
              \@expl@@@shipout@add@foreground@box@@n

\let \AtBeginShipoutUpperLeft
              \@expl@@@shipout@add@background@picture@@n
\let \AtBeginShipoutUpperLeftForeground
              \@expl@@@shipout@add@foreground@picture@@n
```

(*End definition for* `\AtBeginShipoutAddToBox` *and others. These functions are documented on page* 704.)

`\ShipoutBoxHeight` `\ShipoutBoxWidth` `\ShipoutoBoxDepth` This is somewhat different from the original in atbegshi where `\ShipoutBoxHeight` etc. only holds the `\the\ht<box>` value. This may has some implications in some use cases and if that is a problem then it might need changing.

```
\ExplSyntaxOn
\cs_new:Npn \ShipoutBoxHeight { \dim_use:N \l_shipout_box_ht_dim }
\cs_new:Npn \ShipoutBoxDepth  { \dim_use:N \l_shipout_box_dp_dim }
\cs_new:Npn \ShipoutBoxWidth  { \dim_use:N \l_shipout_box_wd_dim }
\ExplSyntaxOff
```

(*End definition for* `\ShipoutBoxHeight`, `\ShipoutBoxWidth`, *and* `\ShipoutoBoxDepth`. *These functions are documented on page* ??.)

```
⟨/atbegshi-ltx⟩
```

If the package is requested we substitute the one above:

```
⟨*2ekernel⟩
\declare@file@substitution{atbegshi.sty}{atbegshi-ltx.sty}
⟨/2ekernel⟩
```

6.3 Package **everyshi** emulation

```
⟨*everyshi-ltx⟩
\ProvidesPackage{everyshi-ltx}
   [2020/08/17 v1.0a
    Emulation of the original everyshi package^^Jwith kernel methods]
```

\EveryShipout
\AtNextShipout

This package has only two public commands so simulating it is easy:

```
\protected \def \EveryShipout  {\AddToHook{shipout/before}}
\protected \def \AtNextShipout {\AddToHookNext{shipout/before}}
```

(*End definition for* `\EveryShipout` *and* `\AtNextShipout`*. These functions are documented on page 704.*)

```
%    This is one difference between \pkg{everyshi} and the kernel
%    implementation, the latter does not directly use box 255.
%
%    For usage by ordinary users this makes no difference but of a
%    package use complicated code together with \pkg{everyshi} and
%    directly manipulates box 255 then this package needs updating.
%    In most cases the updates are simple because the kernel offers
%    hooks that makes such complicated code unnecessary.
%
%    We therefore add a little file into the adjusted package
%    \begin{macrocode}
%%
%%   In normal circumstances the above emulation is sufficient and in
%%   all known packages (we know of) that use everyshi it either works or
%%   the packages have been adjusted.
%%
%%   Code that directly manipulates box 255, however, might fail.
%%   If that is the case look at the shipout hooks offered now as
%%   they are normally sufficienct to avoid such minpulations (or
%%   replace box 255 with \ShipoutBox in the code.
%%
⟨/everyshi-ltx⟩
```

If the package is requested we substitute the one above:

```
⟨*2ekernel⟩
\declare@file@substitution{everyshi.sty}{everyshi-ltx.sty}
⟨/2ekernel⟩
```

Rather important :-)

```
⟨@@=⟩
```

File S
ltoutput.dtx

1 Output Routine

1.1 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
```

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

```
⟨*2ekernel⟩
\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 includes a !
5	1 iff a type 1 float

```
 6       1 iff a type 2 float
 etc.
```

A negative float specifier is used to indicate a marginal note.

MACROS AND DATA STRUCTURES FOR PROCESSING FLOATS

```
A FLOAT LIST consisting of the floats in boxes \boxa ... \boxN has
the form:
        \@elt \boxa ... \@elt \boxN
where   \boxI is defined by
        \newinsert\boxI
Normally, \@elt is \let to \relax.  A test can be performed on the
entire float list by locally \def'ing \@elt appropriately and
executing the list.
This is a lot more efficient than looping through the list.

The following macros are used for manipulating float lists.

\@next \CS \LIST {NONEMPTY}{EMPTY} ==  %% NOTE: ASSUME \@elt
= \relax
   BEGIN  assume that \LIST == \@elt \B1 ... \@elt \Bn
          if n = 0
            then  EMPTY
            else  \CS     :=L \B1
                  \LIST  :=G \@elt \B2 ... \@elt \Bn
                  NONEMPTY
          fi
   END


\@bitor\NUM\LIST : Globally sets switch @test to the disjunction for
        all I of bit  log2 \NUM of the float specifiers of all the
        floats in \LIST.
        I.e., @test is set to true iff there is at least one
        float in \LIST having bit  log2 \NUM  of its float specifier
        equal to 1.

Note: 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.
End of historical LaTeX 2.09 comments.

```
\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}}

\def\@testfalse{\global\let\if@test\iffalse}
\def\@testtrue {\global\let\if@test\iftrue}
\@testfalse

\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.

```
\def\@xbitor #1{\@tempcntb \count#1
   \ifnum \@tempcnta =\z@
   \else
     \divide\@tempcntb\@tempcnta
     \ifodd\@tempcntb \@testtrue\fi
   \fi}
```

DEFINITION OF FLOAT BOXES:

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/10/01}%
⟨latexrelease⟩                  {\bx@ZZ}{Extended float list}%
⟨*2ekernel | latexrelease⟩
\let\@elt\newinsert
⟨*2ekernel⟩
\def\@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}
\@freelist
⟨/2ekernel⟩
\ifx\numexpr\@undefined\else
\def\reserved@a{%
  \@elt\bx@S\@elt\bx@T\@elt\bx@U\@elt\bx@V
  \@elt\bx@W\@elt\bx@X\@elt\bx@Y\@elt\bx@Z
  \@elt\bx@AA\@elt\bx@BB\@elt\bx@CC\@elt\bx@DD\@elt\bx@EE
  \@elt\bx@FF\@elt\bx@GG\@elt\bx@HH\@elt\bx@II\@elt\bx@JJ
  \@elt\bx@KK\@elt\bx@LL\@elt\bx@MM\@elt\bx@NN
  \@elt\bx@OO\@elt\bx@PP\@elt\bx@QQ\@elt\bx@RR
  \@elt\bx@SS\@elt\bx@TT\@elt\bx@UU\@elt\bx@VV
  \@elt\bx@WW\@elt\bx@XX\@elt\bx@YY\@elt\bx@ZZ}
\reserved@a
\def\@elt{\noexpand\@elt\noexpand}
\edef\@freelist{\@freelist\reserved@a}
\fi
\let\reserved@a\relax
\let\@elt\relax
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\bx@ZZ}{Extended float list}%
⟨latexrelease⟩\def\@freelist{%
⟨latexrelease⟩  \@elt\bx@A\@elt\bx@B\@elt\bx@C\@elt\bx@D\@elt\bx@E
⟨latexrelease⟩  \@elt\bx@F\@elt\bx@G\@elt\bx@H\@elt\bx@I\@elt\bx@J
⟨latexrelease⟩  \@elt\bx@K\@elt\bx@L\@elt\bx@M\@elt\bx@N
⟨latexrelease⟩  \@elt\bx@O\@elt\bx@P\@elt\bx@Q\@elt\bx@R}
⟨latexrelease⟩  \insc@unt=234
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

```
\gdef\@toplist{}
\gdef\@botlist{}
\gdef\@midlist{}
\gdef\@currlist{}
\gdef\@deferlist{}
\gdef\@dbltoplist{}
```

The new algorithm stores page wide floats together with column floats in a single `\@deferlist` list. We keep `\@dbldeferlist` initialised as empty so that packages that are testing for deferred floats can use the same code for old or new float handling.

```
\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.

This interface is no longer used. Instead a new one is inside ltshipout.dtx. We only keep the box in case some old code refers to it directly (or we do some rollback).

```
\newbox\@begindvibox
%\DeclareRobustCommand \AtBeginDvi [1]{%
%  \global \setbox \@begindvibox
%    \vbox{\unvbox \@begindvibox #1}%
%}
```

(*End definition for* \AtBeginDvi *and* \@begindvibox*. These functions are documented on page* 702.)

\@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
```

(*End definition for* \@maxdepth*. This function is documented on page* ??.)

\paperheight
\paperwidth

New \paper... registers.

```
\newdimen\paperheight
\newdimen\paperwidth
```

(*End definition for* \paperheight *and* \paperwidth*. These functions are documented on page* ??.)

\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 document 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
```

(*End definition for* `\if@insert` *and others. These functions are documented on page* **??**.)

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

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}
```

End of historical LaTeX 2.09 comments.

`\clearpage` The tests at the beginning are an experimental attempt to avoid a completely empty page after a `\twocolumn[...]`. This prevents the text from the argument vanishing into a float box, never to be seen again. We hope that it does not produce wrong formatting in other cases.

```
\def\clearpage{%
  \ifvmode
    \ifnum \@dbltopnum =\m@ne
      \ifdim \pagetotal <\topskip
        \hbox{}%
      \fi
    \fi
  \fi
  \newpage
  \write\m@ne{}%
  \vbox{}%
  \penalty -\@Mi
}
```

(*End definition for* `\clearpage`*. This function is documented on page* **??**.)

\cleardoublepage

```
\def\cleardoublepage{\clearpage\if@twoside \ifodd\c@page\else
    \hbox{}\newpage\if@twocolumn\hbox{}\newpage\fi\fi\fi}
⟨/2ekernel⟩
```

(*End definition for* `\cleardoublepage`*. This function is documented on page* **??**.)

\onecolumn

```
⟨*2ekernel | fltrace⟩
\def\onecolumn{%
  \clearpage
  \global\columnwidth\textwidth
  \global\hsize\columnwidth
  \global\linewidth\columnwidth
  \global\@twocolumnfalse
  \col@number \@ne
  \@floatplacement}
```

(*End definition for* `\onecolumn`*. This function is documented on page* **??**.)

\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`.

```
⟨/2ekernel | fltrace⟩
⟨latexrelease⟩\IncludeInRelease{2017/04/15}%
⟨latexrelease⟩                    {\newpage}{Check depth of page}%
⟨*2ekernel | latexrelease | fltrace⟩
\def \newpage {%
  \if@noskipsec
    \ifx \@nodocument\relax
      \leavevmode
      \global \@noskipsecfalse
    \fi
  \fi
  \if@inlabel
    \leavevmode
    \global \@inlabelfalse
  \fi
  \if@nobreak \@nobreakfalse \everypar{}\fi
  \par
```

The `\vfil` at the end of the macro before the break penalty will normally result in the page being run short, even with `\flushbottom` in effect (in contrast to the behavior of `\pagebreak`). However, if there is some explicit stretch on the page, say, a `\vfill`, it has the undesired side-effect, that the last line will not align at its baseline if it contains characters going below the baseline, as the value of `\prevdepth` is no longer taken into account by TEX. So we back up by that amount (or by `\maxdepth` if it is really huge), to mimic the normal behavior without the `\newpage`.

```
  \ifdim\prevdepth>\z@
     \vskip -%
       \ifdim\prevdepth>\maxdepth
```

```
        \maxdepth
      \else
        \prevdepth
      \fi
  \fi
  \vfil
  \penalty -\@M}
⟨/2ekernel | latexrelease | fltrace⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\newpage}{Check depth of page}%
⟨latexrelease⟩\def \newpage {%
⟨latexrelease⟩  \if@noskipsec
⟨latexrelease⟩    \ifx \@nodocument\relax
⟨latexrelease⟩      \leavevmode
⟨latexrelease⟩      \global \@noskipsecfalse
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@inlabel
⟨latexrelease⟩    \leavevmode
⟨latexrelease⟩    \global \@inlabelfalse
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@nobreak \@nobreakfalse \everypar{}\fi
⟨latexrelease⟩  \par
⟨latexrelease⟩  \vfil
⟨latexrelease⟩  \penalty -\@M}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel | fltrace⟩
```

(*End definition for* `\newpage`*. This function is documented on page* **??**.)

`\@emptycol` It may be better to use an invisible rule rather than an empty box here.

```
\def \@emptycol {\vbox{}\penalty -\@M}
```

(*End definition for* `\@emptycol`*. This function is documented on page* **??**.)

`\twocolumn` `\@topnewpage` There are several bug fixes to the two-column stuff here.

```
\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⟩
    \fl@trace{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
}
```

(*End definition for* `\twocolumn` *and* `\@topnewpage`*. These functions are documented on page* **??**.)

`\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⟩
       \fl@trace{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
}
```

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

CHANGES TO `\@specialoutput`:

* `\penalty\z@` changed to `\penalty\interlinepenalty` so `\samepage` works properly with figure and table environments.
  (Changed 23 Oct 86)

* Definition of `\@specialoutput` changed 26 Feb 88 so `\@pageht` and `\@pagedp` aren't changed for a marginal note.
  (Change suggested by Chris Rowley.)

End of historical LaTeX 2.09 comments.

```
\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
          \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⟩
              \fl@trace {Extra size added: \the \ht\@kludgeins}%
⟨/trace⟩
            \fi
          \fi
```

This version puts the inserts back just before the additional material; it could be moved earlier, before unboxing the page-so-far. Neither is guaranteed not to put things on the wrong page. This version is similar to the original version.

```
            \@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 | fltrace⟩
```

(*End definition for* `\output` *and* `\@specialoutput`*. These functions are documented on page* **??**.)

`\@testwrongwidth` `\f@depth`

Test if the float box has the wrong width when trying to place it into some area. (Actually the test is for a conventional depth setting rather than for the width of the float. For that reason the box depth was explicitly tailored when the float was created).

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\@testwrongwidth}{float order in 2-column}%
⟨*2ekernel | latexrelease | fltrace⟩
```

```
\def\@testwrongwidth #1{%
  \ifdim\dp#1=\f@depth
⟨*trace⟩
    \fl@trace{\string#1
              \ifdim\f@depth=\z@ single \else double \fi
              column float -- ok}%
⟨/trace⟩
  \else
    \global\@testtrue
⟨*trace⟩
    \fl@trace{\string#1
              \ifdim\f@depth=\z@ double \else single \fi
              column float -- wrong}%
⟨/trace⟩
  \fi}%
```

Normally looking for single column floats, which have zero depth.

```
\let\f@depth\z@
```

```
⟨/2ekernel | latexrelease | fltrace⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\@testwrongwidth}{float order in 2-column}%
⟨latexrelease⟩\let\@testwrongwidth\@undefined
⟨latexrelease⟩\let\f@depth\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\@testwrongwidth` *and* `\f@depth`*. These functions are documented on page* **??**.)

`\@doclearpage` This is a very much an emergency action, just dumping everything: footnotes first then floats. A more sophisticated version is needed; but even more urgent is a bug-free version (see, for example, pr/3528).

Also, it puts any left-over non-boxes (writes, specials, etc.) back after any float pages created: this is a very bad bug since, for example, a kludge insert will be in quite the wrong place and, worse, be irremovable and uncancelable.

All the remaining changes are replacing the double column defer list or inserting the extra test `\@testwrongwidth{`⟨*box*⟩`}` at suitable places. That is at places where a box is taken off the deferlist.

```
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@doclearpage}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨*2ekernel | latexrelease⟩
\def \@doclearpage {%
     \ifvoid\footins
       \ifvbox\@kludgeins
         {\setbox \@tempboxa \box \@kludgeins}%
⟨*trace⟩
         \fl@trace {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\@deferlist{\@dbltoplist\@deferlist}%
           \global \let \@dbltoplist \@empty
           \global \@colht \textheight
           \begingroup
              \@dblfloatplacement
```

```
              \@makefcolumn\@deferlist
              \@whilesw\if@fcolmade \fi{\@outputpage
                                        \@makefcolumn\@deferlist}%
            \endgroup
          \else
            \vbox{}\clearpage
          \fi
        \fi
```

the next line is needed to avoid losing floats in certain circumstances a single call to the original `\doclearpage` will now no longer output all floats.

```
        \ifx\@deferlist\@empty \else\clearpage \fi
      \else
        \setbox\@cclv\vbox{\box\@cclv\vfil}%
        \@makecol\@opcol
        \clearpage
      \fi
}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@doclearpage}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨latexrelease⟩\def \@doclearpage {%
⟨latexrelease⟩      \ifvoid\footins
```

We empty any left over kludge insert box here; this is a temporary fix. It should perhaps be applied to one page of cleared floats, but who cares? The whole of this stuff needs completely redoing for many such reasons.

```
⟨latexrelease⟩        \ifvbox\@kludgeins
⟨latexrelease⟩          {\setbox \@tempboxa \box \@kludgeins}%
⟨*trace⟩
⟨latexrelease⟩          \fl@trace {kludgeins box made void}%
⟨/trace⟩
⟨latexrelease⟩        \fi
⟨latexrelease⟩        \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
⟨latexrelease⟩        \setbox\@tempboxa\box\@cclv
⟨latexrelease⟩        \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
⟨latexrelease⟩        \global \let \@toplist \@empty
⟨latexrelease⟩        \global \let \@botlist \@empty
⟨latexrelease⟩        \global \@colroom \@colht
⟨latexrelease⟩        \ifx \@currlist\@empty
⟨latexrelease⟩        \else
⟨latexrelease⟩           \@latexerr{Float(s) lost}\@ehb
⟨latexrelease⟩           \global \let \@currlist \@empty
⟨latexrelease⟩        \fi
⟨latexrelease⟩        \@makefcolumn\@deferlist
⟨latexrelease⟩        \@whilesw\if@fcolmade \fi
⟨latexrelease⟩                      {\@opcol\@makefcolumn\@deferlist}%
⟨latexrelease⟩        \if@twocolumn
⟨latexrelease⟩          \if@firstcolumn
⟨latexrelease⟩            \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
```

```
⟨latexrelease⟩            \global \let \@dbltoplist \@empty
⟨latexrelease⟩            \global \@colht \textheight
⟨latexrelease⟩            \begingroup
⟨latexrelease⟩               \@dblfloatplacement
⟨latexrelease⟩               \@makefcolumn\@dbldeferlist
⟨latexrelease⟩               \@whilesw\if@fcolmade \fi
⟨latexrelease⟩                     {\@outputpage\@makefcolumn\@dbldeferlist}%
⟨latexrelease⟩            \endgroup
⟨latexrelease⟩          \else
⟨latexrelease⟩            \vbox{}\clearpage
⟨latexrelease⟩          \fi
⟨latexrelease⟩        \fi
⟨latexrelease⟩      \else
⟨latexrelease⟩        \setbox\@cclv\vbox{\box\@cclv\vfil}%
⟨latexrelease⟩        \@makecol\@opcol
⟨latexrelease⟩        \clearpage
⟨latexrelease⟩      \fi
⟨latexrelease⟩   }%
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\@doclearpage`*. This function is documented on page* **??**.)

`\@opcol` Several changes in detail here.

```
⟨*2ekernel | fltrace⟩
\def \@opcol {%
  \if@twocolumn
    \@outputdblcol
  \else
    \@outputpage
⟨*trace⟩
    \fl@trace{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 | fltrace⟩
```

(*End definition for* `\@opcol`*. This function is documented on page* **??**.)

`\@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⟩
\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.

```
  \ifvbox\@kludgeins
    \@makespecialcolbox
  \else
```

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 adjustment should be done again after it.

I think that the final boxing of the main text page could have a common ending which may make it simpler to see what is going on.

This needs further investigation, especially in the 'special case'.

Also, the \boxmaxdepth setting here affects what happens within \@texttop and \@textbottom, should it? Is it needed at all?

RmS 91/10/22: Replaced \dimen128 by \dimen@.

```
     \setbox\@outputbox \vbox to\@colht {%
%       \boxmaxdepth \maxdepth                    %??
      \@texttop
      \dimen@ \dp\@outputbox
      \unvbox \@outputbox
```

```
      \vskip -\dimen@
      \@textbottom
      }%
  \fi
  \global \maxdepth \@maxdepth
}
```

(*End definition for* `\@makecol`*. This function is documented on page* **??**.)

`\@reinserts` This is the code which reinserts the inserts. It puts them all in one place; this can make some of them come out on the wrong page. It has been put into a separate macro to expedite experimentation.

```
\gdef \@reinserts{%
  \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
  \ifvbox\@kludgeins\insert\@kludgeins
                      {\unvbox\@kludgeins}\fi
}
⟨/2ekernel⟩
```

(*End definition for* `\@reinserts`*. This function is documented on page* **??**.)

`\@makespecialcolbox` This implements certain variations in page-makeup.

```
⟨*2ekernel | fltrace⟩
\gdef \@makespecialcolbox {%
⟨*trace⟩
  \fl@trace{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⟩
    \fl@trace {Natural ht of col: \the \ht\@outputbox}%
    \fl@trace {\string \@colht: \the \@colht}%
    \fl@trace {Pageshrink added: \the \pageshrink}%
    \fl@trace {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⟩
    \fl@trace {Natural ht of col: \the \ht\@outputbox}%
    \fl@trace {\string \@colht: \the \@colht}%
    \fl@trace {Extra size added: -\the \ht \@kludgeins}%
    \fl@trace {Hence, height of inner box: \the \@tempdima}%
    \fl@trace {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⟩
    \fl@trace {kludgeins box made void}%
⟨/trace⟩
}
⟨/2ekernel | fltrace⟩
```

(*End definition for* `\@makespecialcolbox`. *This function is documented on page* **??**.)

\@texttop
\@textbottom

These do nothing as a default.

```
⟨*2ekernel⟩
\let \@texttop \relax
\let \@textbottom \relax
```

(*End definition for* `\@texttop` *and* `\@textbottom`*. These functions are documented on page* **??**.)

\@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
```

(*End definition for* `\@resetactivechars` *and* `\@activechar@info`*. These functions are documented on page* **??**.)

\@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 to prevent mis-use of internal commands as hooks.

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2017/04/15}%
⟨latexrelease⟩  {\@outputpage}{Reset language for hyphenation}%
⟨*2ekernel | latexrelease⟩
\def\@outputpage{%
```

The \endgroup is put in by \aftergroup.

```
\begingroup
```

Now all the set-up stuff has been in-lined for Frank.

First the stuff for the writes.

From here ... was in the command \@writesetup.

```
 \let \protect \noexpand
```

RmS 93/08/19: Redefined accents to allow changes in font encoding; but exactly why was this needed?

Reset \language to the value current at \begin{document}. In particular this ensures that a pagebreak in verbatim does not prevent hyphenation in the page head.

```
 \language\document@default@language
```

The \catcode`\ = 10 was removed as it was considered useless (presumably because nothing gets tokenised during shipout).

This was put in as some error produced active spaces in a mark, I think.

Why was the hyphen reset?

```
 \@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
```

Correct? or just restore by ending the group?

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

This first bit has been moved inside the shipped out box.

Now the setup inside the shipped out box; this should contain all the stuff that could only affect typesetting; other stuff may need to be reset for the writes also.

From here ... was in the command `\@shipoutsetup`.

```
\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
}
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\@outputpage}{Reset language for hyphenation}%
⟨latexrelease⟩\def\@outputpage{%
⟨latexrelease⟩\begingroup
⟨latexrelease⟩  \let \protect \noexpand
⟨latexrelease⟩  \@resetactivechars
⟨latexrelease⟩  \global\let\@@if@newlist\if@newlist
⟨latexrelease⟩  \global\@newlistfalse
⟨latexrelease⟩  \@parboxrestore
⟨latexrelease⟩  \shipout \vbox{%
⟨latexrelease⟩    \set@typeset@protect
⟨latexrelease⟩    \aftergroup \endgroup
⟨latexrelease⟩    \aftergroup \set@typeset@protect
⟨latexrelease⟩  \if@specialpage
⟨latexrelease⟩    \global\@specialpagefalse\@nameuse{ps@\@specialstyle}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@twoside
⟨latexrelease⟩    \ifodd\count\z@
⟨latexrelease⟩         \let\@thehead\@oddhead \let\@thefoot\@oddfoot
⟨latexrelease⟩         \let\@themargin\oddsidemargin
⟨latexrelease⟩    \else \let\@thehead\@evenhead
⟨latexrelease⟩       \let\@thefoot\@evenfoot \let\@themargin\evensidemargin
⟨latexrelease⟩    \fi
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \reset@font
⟨latexrelease⟩  \normalsize
⟨latexrelease⟩  \normalsfcodes
⟨latexrelease⟩  \let\label\@gobble
⟨latexrelease⟩  \let\index\@gobble
⟨latexrelease⟩  \let\glossary\@gobble
⟨latexrelease⟩  \baselineskip\z@skip \lineskip\z@skip \lineskiplimit\z@
⟨latexrelease⟩    \@begindvi
⟨latexrelease⟩    \vskip \topmargin
⟨latexrelease⟩    \moveright\@themargin \vbox {%
⟨latexrelease⟩      \setbox\@tempboxa \vbox to\headheight{%
⟨latexrelease⟩        \vfil
⟨latexrelease⟩        \color@hbox
⟨latexrelease⟩          \normalcolor
```

```
⟨latexrelease⟩          \hb@xt@\textwidth{\@thehead}%
⟨latexrelease⟩        \color@endbox
⟨latexrelease⟩        }%
⟨latexrelease⟩      \dp\@tempboxa \z@
⟨latexrelease⟩      \box\@tempboxa
⟨latexrelease⟩      \vskip \headsep
⟨latexrelease⟩      \box\@outputbox
⟨latexrelease⟩      \baselineskip \footskip
⟨latexrelease⟩      \color@hbox
⟨latexrelease⟩        \normalcolor
⟨latexrelease⟩        \hb@xt@\textwidth{\@thefoot}%
⟨latexrelease⟩      \color@endbox
⟨latexrelease⟩      }%
⟨latexrelease⟩    }%
⟨latexrelease⟩  \global\let\if@newlist\@@if@newlist
⟨latexrelease⟩  \global \@colht \textheight
⟨latexrelease⟩  \stepcounter{page}%
⟨latexrelease⟩  \let\firstmark\botmark
⟨latexrelease⟩}
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@outputpage`, `\@shipoutsetup`, *and* `\@writesetup`. *These functions are documented on page* ??.)

`\@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
}
```

(*End definition for* `\@begindvi`. *This function is documented on page* ??.)

`\@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 pseudocode.

```
\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
}
```

(*End definition for* `\@combinefloats`, `\@cflt`, *and* `\@cflb`. *These functions are documented on page* ??.)

`\@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
```

If pdf links are present in the galley and those links get broken across pages they have to end up being on the same level of boxing (even if not actually in the same structure) due to some engine restrictions in pdfTEX and LuaTEX. We therefore unbox `\@outputbox` here (which only contains a single `\hbox`) so that this case has the same boxing level as a normal twocolumn page without top floats.

```
        \unvbox\@outputbox
        }%
  \fi
}
⟨/2ekernel⟩
```

(*End definition for* `\@comflelt`, `\@comdblflelt`, *and* `\@combinedblfloats`. *These functions are documented on page* ??.)

`\@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 | fltrace⟩
\def \@startcolumn {%
  \global \@colroom \@colht
  \@tryfcolumn \@deferlist
  \if@fcolmade
⟨*trace⟩
    \fl@trace{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`.

```
⟨/2ekernel | fltrace⟩
⟨latexrelease | fltrace⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease | fltrace⟩  {\@startdblcolumn}{float order in 2-column}%
⟨*2ekernel | latexrelease | fltrace⟩
\def \@startdblcolumn {%
  \@tryfcolumn \@deferlist
  \if@fcolmade
⟨fltrace⟩    \fl@trace{PAGE: double float page completed}%
  \else
    \begingroup
      \let \reserved@b \@deferlist
      \global \let \@deferlist \@empty
      \let \@elt \@sdblcolelt
      \reserved@b
    \endgroup
```

```
  \fi
}%
⟨/2ekernel | latexrelease | fltrace⟩
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨latexrelease | fltrace⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace⟩  {\@startdblcolumn}{float order in 2-column}%
⟨latexrelease | fltrace⟩\def \@startdblcolumn {%
```

Not needed since this always comes after `\@outputpage`:

```
⟨latexrelease | fltrace⟩% \global \@colht \textheight
⟨latexrelease | fltrace⟩  \@tryfcolumn \@dbldeferlist
⟨latexrelease | fltrace⟩  \if@fcolmade
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{PAGE: double float page completed}%
⟨/trace⟩
⟨latexrelease | fltrace⟩  \else
```

```
⟨latexrelease | fltrace⟩    \begingroup
⟨latexrelease | fltrace⟩      \let \reserved@b \@dbldeferlist
⟨latexrelease | fltrace⟩      \global \let \@dbldeferlist \@empty
⟨latexrelease | fltrace⟩      \let \@elt \@sdblcolelt
⟨latexrelease | fltrace⟩      \reserved@b
⟨latexrelease | fltrace⟩    \endgroup
⟨latexrelease | fltrace⟩  \fi
⟨latexrelease | fltrace⟩}%
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨*2ekernel | fltrace⟩
```

(*End definition for* `\@startcolumn` *and* `\@startdblcolumn`*. These functions are documented on page* ??.)

`\@tryfcolumn` Now tests if its list is empty before any further exertion.

```
\def \@tryfcolumn #1{%
  \global \@fcolmadefalse
  \ifx #1\@empty
  \else
⟨*trace⟩
      \fl@trace{PAGE: try float \if@twocolumn column/page\else page\fi
                   ---\string #1}%
      \fl@trace{----- \string #1: #1}%
⟨/trace⟩
```

```
    \xdef\@trylist{#1}%
    \global \let \@failedlist \@empty
    \begingroup
      \let \@elt \@xtryfc \@trylist
    \endgroup
    \if@fcolmade
      \@vtryfc #1%
    \fi
  \fi
}
⟨/2ekernel | fltrace⟩
```

(*End definition for* `\@tryfcolumn`*. This function is documented on page* ??.)

```
⟨*2ekernel⟩
```

\@scolelt

```
\def\@scolelt#1{\def\@currbox{#1}\@addtonextcol}
```

(*End definition for* `\@scolelt`*. This function is documented on page* **??**.)

\@sdblcolelt

```
\def\@sdblcolelt#1{\def\@currbox{#1}\@addtodblcol}
```

(*End definition for* `\@sdblcolelt`*. This function is documented on page* **??**.)

\@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}}
```

(*End definition for* `\@vtryfc`*. This function is documented on page* **??**.)

\@wtryfc

```
\def\@wtryfc #1{%
  \global\setbox\@outputbox\vbox{%
    \unvbox\@outputbox
    \vskip\@fpsep
    \box #1}}
```

(*End definition for* `\@wtryfc`*. This function is documented on page* **??**.)

\@xtryfc

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{\@xtryfc}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨*2ekernel | latexrelease⟩
\def\@xtryfc #1{%
  \@next\reserved@a\@trylist{}{}%
  \@currtype \count #1%
  \divide\@currtype\@xxxii
  \multiply\@currtype\@xxxii
  \@bitor \@currtype \@failedlist
  \@testfp #1%
  \@testwrongwidth #1%
  \ifdim \ht #1>\@colht
     \@testtrue
  \fi
  \if@test
    \@cons\@failedlist #1%
  \else
```

```
    \@ytryfc #1%
  \fi}%
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{\@xtryfc}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨latexrelease⟩\def\@xtryfc #1{%
⟨latexrelease⟩  \@next\reserved@a\@trylist{}{}%
⟨latexrelease⟩  \@currtype \count #1%
⟨latexrelease⟩  \divide\@currtype\@xxxii
⟨latexrelease⟩  \multiply\@currtype\@xxxii
⟨latexrelease⟩  \@bitor \@currtype \@failedlist
⟨latexrelease⟩  \@testfp #1%
⟨latexrelease⟩  \ifdim \ht #1>\@colht
⟨latexrelease⟩    \@testtrue
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@test
⟨latexrelease⟩    \@cons\@failedlist #1%
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@ytryfc #1%
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\@xtryfc`*. This function is documented on page* **??**.)

`\@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}
```

(*End definition for* `\@ytryfc`*. This function is documented on page* **??**.)

`\@ztryfc`

```
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}{@ztryfc}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨*2ekernel | latexrelease⟩
\def\@ztryfc #1{%
  \@tempcnta\count #1%
  \divide\@tempcnta\@xxxii
  \multiply\@tempcnta\@xxxii
```

```
  \@bitor \@tempcnta {\@failedlist \@flfail}%
  \@testfp #1%
```

not in fixfloats?

```
  \@testwrongwidth #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 | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}{@ztryfc}%
⟨latexrelease⟩                              {float order in 2-column}%
⟨latexrelease⟩\def\@ztryfc #1{%
⟨latexrelease⟩  \@tempcnta \count#1%
⟨latexrelease⟩  \divide\@tempcnta\@xxxii
⟨latexrelease⟩  \multiply\@tempcnta\@xxxii
⟨latexrelease⟩  \@bitor \@tempcnta {\@failedlist \@flfail}%
⟨latexrelease⟩  \@testfp #1%
⟨latexrelease⟩  \@tempdimb\@tempdima
⟨latexrelease⟩  \advance\@tempdimb \ht#1%
⟨latexrelease⟩  \advance\@tempdimb\@fpsep
⟨latexrelease⟩  \ifdim \@tempdimb >\@colht
⟨latexrelease⟩    \@testtrue
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \if@test
⟨latexrelease⟩    \@cons\@flfail #1%
⟨latexrelease⟩  \else
⟨latexrelease⟩    \@cons\@flsucceed #1%
⟨latexrelease⟩    \@tempdima\@tempdimb
⟨latexrelease⟩  \fi}%
⟨latexrelease⟩\EndIncludeInRelease
```

(*End definition for* `\@ztryfc`*. This function is documented on page* **??**.)

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 | fltrace⟩
\def \@addtobot {%
⟨*trace⟩
   \fl@trace{***Start addtobot}%
⟨/trace⟩
   \@getfpsbit 4\relax
⟨*trace⟩
   \fl@trace{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⟩
         \fl@trace{colroom (after-bot) = \the \@colroom}%
         \fl@trace{colnum (after-bot) = \the \@colnum}%
         \fl@trace{botnum (after-bot) = \the \@botnum}%
         \fl@trace{***Success: bot}%
⟨/trace⟩
         \@inserttrue
       \fi
⟨*trace⟩
     \else
       \fl@trace{Fail: botnum = \the \@botnum:
                                 fpstype \the \@fpstype=ORD?}%
       \ifnum \@fpstype<\sixt@@n
         \fl@trace{ERROR: !b float not successful (addtobot)}%
       \fi
⟨/trace⟩
     \fi
   \fi
}
```

(*End definition for* `\@addtobot`*. This function is documented on page* **??**.)

`\@addtotoporbot` Lots of changes.

```
\def \@addtotoporbot {%
⟨*trace⟩
   \fl@trace{***Start addtotoporbot}%
⟨/trace⟩
   \@getfpsbit \tw@
⟨*trace⟩
   \fl@trace{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⟩
           \fl@trace{(mid+bot)list: \@midlist, \@botlist:
```

```
                                (addtotoporbot-before)}%
⟨/trace⟩
          \if@test
⟨*trace⟩
          \fl@trace{type already on list: mid or bot---sent to addtobot}%
⟨/trace⟩
          \else
           \@flupdates \@topnum \@toproom \@toplist
⟨*trace⟩
           \fl@trace{colroom (after-top) = \the \@colroom}%
           \fl@trace{colnum (after-top) = \the \@colnum}%
           \fl@trace{topnum (after-top) = \the \@topnum}%
           \fl@trace{***Success: top}%
⟨/trace⟩
           \@inserttrue
          \fi
        \fi
⟨*trace⟩
      \else
        \fl@trace{Fail: topnum = \the \@topnum: fpstype
                                            \the \@fpstype=ORD?}%
        \ifnum \@fpstype<\sixt@@n
          \fl@trace{ERROR: !t float not successful (addtotoporbot)}%
        \fi
⟨/trace⟩
      \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
      \fl@trace{sent to addtobot (addtotoporbot)}%
⟨/trace⟩
      \@addtobot
   \fi
}
⟨/2ekernel | fltrace⟩
```

(*End definition for* `\@addtotoporbot`*. This function is documented on page* ??.)

`\@addtocurcol` Lots of changes.

```
⟨latexrelease | fltrace | flafter⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease | fltrace | flafter⟩  {\@addtocurcol}{float order in 2-column}%
⟨*2ekernel | latexrelease | fltrace | flafter⟩
\def \@addtocurcol {%
⟨*trace⟩
  \fl@trace{***Start addtocurcol}%
⟨/trace⟩
   \@insertfalse
   \@setfloattypecounts
   \ifnum \@fpstype=8
⟨*trace⟩
      \fl@trace{fpstype !p only (addtocurcol): \the \@fpstype = 8?}%
⟨/trace⟩
   \else
```

```
    \ifnum \@fpstype=24
⟨*trace⟩
      \fl@trace{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⟩
      \fl@trace{textfloatsheight (before) = \the \@textfloatsheight}%
⟨/trace⟩
      \advance \@textmin \@textfloatsheight
      \@reqcolroom \@pageht
```

This line must be removed since `\@specialoutput` changed.

```
%       \advance \@reqcolroom \@pagedp
⟨*trace⟩
      \fl@trace{textmin + textfloatsheight: \the \@textmin}%
      \fl@trace{page-so-far: \the \@reqcolroom}%
⟨/trace⟩
      \ifdim \@textmin>\@reqcolroom
        \@reqcolroom \@textmin
⟨*trace⟩
        \fl@trace{ORD? textmin being used}%
⟨/trace⟩
      \fi
      \advance \@reqcolroom \ht\@currbox
⟨*trace⟩
      \fl@trace{float size = \the \ht \@currbox (addtocurcol)}%
      \fl@trace{colroom = \the \@colroom (addtocurcol)}%
      \fl@trace{reqcolroom = \the \@reqcolroom (addtocurcol)}%
⟨/trace⟩
      \ifdim \@colroom>\@reqcolroom
        \@flsetnum \@colnum
        \ifnum \@colnum>\z@
          \@bitor\@currtype\@deferlist
```

We need to defer the float also if its width doesn't fit.

```
          \@testwrongwidth\@currbox
⟨*trace⟩
          \fl@trace{deferlist: \@deferlist: (addtocurcol-before)}%
⟨/trace⟩
          \if@test
⟨*trace⟩
            \fl@trace{type already on list: defer (addtocurcol)}%
⟨/trace⟩
          \else
            \@bitor\@currtype\@botlist
⟨*trace⟩
          \fl@trace{botlist: \@botlist: (addtocurcol-before)}%
⟨/trace⟩
```

```
            \if@test
⟨*trace⟩
              \fl@trace{type already on list: bot---sent to addtobot}%
⟨/trace⟩
              \@addtobot
            \else
⟨*trace⟩
              \fl@trace{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⟩
                \fl@trace{***Success: here}%
                \fl@trace{textfloatsheight (after-here) =
                     \the \@textfloatsheight}%
                \fl@trace{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 between paragraphs (i.e. when in vertical mode).

If the nobreak switch is true we need to reset it and clear `\everypar` since the float may not reset the flag and cannot reset the `\everypar` globally.

Typesetting starts here (we are in vertical mode).

```
                  \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
                    \fl@trace{Fail---no room at 2nd test of colroom
                                  (addtocorcol \string\intextsep)}%
⟨/trace⟩
                  \fi
                \fi
                \if@insert
                \else
```

Next set of docstrip guards are a bit weird, essentially `\@addtotoporbot` ends up inside the kernel and the `fltrace` package and `\@addtobot` shows up in the `flafter` package. Guess that could have been done a bit more obvious :-)

```
⟨*2ekernel | fltrace | latexrelease⟩
⟨*trace⟩
                  \fl@trace{not here: sent to addtotoporbot}%
⟨/trace⟩
                  \@addtotoporbot
⟨/2ekernel | fltrace | latexrelease⟩
⟨*!2ekernel&!fltrace&!latexrelease⟩
⟨*trace⟩
                  \fl@trace{not here: sent to addtobot}%
⟨/trace⟩
                  \@addtobot
⟨/!2ekernel&!fltrace&!latexrelease⟩
                \fi
              \fi
            \fi
⟨*trace⟩
          \else
            \fl@trace{Fail: colnum = \the \@colnum:
                        fpstype \the \@fpstype=ORD?}%
            \ifnum \@fpstype<\sixt@@n
              \fl@trace{ERROR: BANG float not successful (addtocurcol)}%
            \fi
⟨/trace⟩
          \fi
⟨*trace⟩
        \else
          \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
                                                  (addtocurcol)}%
⟨/trace⟩
        \fi
      \fi
    \fi
    \if@insert
    \else
      \@resethfps
⟨*trace⟩
      \fl@trace{put on deferlist (addtocurcol)}%
⟨/trace⟩
      \@cons\@deferlist\@currbox
⟨*trace⟩
```

```
    \fl@trace{deferlist: \@deferlist: (addtocurcol-after)}%
⟨/trace⟩
  \fi
}%
⟨/2ekernel | latexrelease | fltrace | flafter⟩
⟨latexrelease | fltrace | flafter⟩\EndIncludeInRelease
⟨latexrelease | fltrace | flafter⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace | flafter⟩  {\@addtocurcol}{float order in 2-column}%
⟨latexrelease | fltrace | flafter⟩\def \@addtocurcol {%
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩  \fl@trace{***Start addtocurcol}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩   \@insertfalse
⟨latexrelease | fltrace | flafter⟩   \@setfloattypecounts
⟨latexrelease | fltrace | flafter⟩   \ifnum \@fpstype=8
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩     \fl@trace{fpstype !p only (addtocurcol):
⟨latexrelease | fltrace | flafter⟩                            \the \@fpstype = 8?}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩   \else
⟨latexrelease | fltrace | flafter⟩     \ifnum \@fpstype=24
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩      \fl@trace{fpstype p only (addtocurcol):
⟨latexrelease | fltrace | flafter⟩                          \the \@fpstype = 24?}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩     \else
⟨latexrelease | fltrace | flafter⟩       \@flsettextmin
```

This is a new adjustment which is quite a major change in functionality; but it implements the documentation. Note that `\@reqcolroom` will include the whole of the page-so-far, and hence includes `\@textfloatsheight` of floats, so before comparing it with `\@textmin`, we add this to `\@textmin` also.

```
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩       \fl@trace{textfloatsheight (before) =
⟨latexrelease | fltrace | flafter⟩                        \the \@textfloatsheight}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩       \advance \@textmin \@textfloatsheight
⟨latexrelease | fltrace | flafter⟩       \@reqcolroom \@pageht
```

This line must be removed since `\@specialoutput` changed.

```
%       \advance \@reqcolroom \@pagedp
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩       \fl@trace{textmin + textfloatsheight:
⟨latexrelease | fltrace | flafter⟩                                 \the \@textmin}%
⟨latexrelease | fltrace | flafter⟩       \fl@trace{page-so-far: \the \@reqcolroom}%
⟨latexrelease | fltrace | flafter⟩
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩       \ifdim \@textmin>\@reqcolroom
⟨latexrelease | fltrace | flafter⟩         \@reqcolroom \@textmin
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩         \fl@trace{ORD? textmin being used}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩       \fi
⟨latexrelease | fltrace | flafter⟩       \advance \@reqcolroom \ht\@currbox
⟨*trace⟩
```

```
⟨latexrelease | fltrace | flafter⟩        \fl@trace{float size =
⟨latexrelease | fltrace | flafter⟩                \the \ht \@currbox (addtocurcol)}%
⟨latexrelease | fltrace | flafter⟩        \fl@trace{colroom =
⟨latexrelease | fltrace | flafter⟩                    \the \@colroom (addtocurcol)}%
⟨latexrelease | fltrace | flafter⟩        \fl@trace{reqcolroom =
⟨latexrelease | fltrace | flafter⟩                 \the \@reqcolroom (addtocurcol)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩        \ifdim \@colroom>\@reqcolroom
⟨latexrelease | fltrace | flafter⟩          \@flsetnum \@colnum
⟨latexrelease | fltrace | flafter⟩          \ifnum \@colnum>\z@
⟨latexrelease | fltrace | flafter⟩            \@bitor\@currtype\@deferlist
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩            \fl@trace{deferlist:
⟨latexrelease | fltrace | flafter⟩               \@deferlist: (addtocurcol-before)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩            \if@test
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩              \fl@trace{type already on list:
⟨latexrelease | fltrace | flafter⟩                             defer (addtocurcol)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩            \else
⟨latexrelease | fltrace | flafter⟩              \@bitor\@currtype\@botlist
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩            \fl@trace{botlist: \@botlist:
⟨latexrelease | fltrace | flafter⟩                            (addtocurcol-before)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩              \if@test
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩                \fl@trace{type already on list:
⟨latexrelease | fltrace | flafter⟩                          bot---sent to addtobot}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩                \@addtobot
⟨latexrelease | fltrace | flafter⟩              \else
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩                \fl@trace{fpstype
⟨latexrelease | fltrace | flafter⟩                \ifodd \@tempcnta OK \else not \fi
⟨latexrelease | fltrace | flafter⟩                here: \the \@fpstype}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩                \ifodd \count\@currbox
⟨latexrelease | fltrace | flafter⟩                  \advance \@reqcolroom \intextsep
⟨latexrelease | fltrace | flafter⟩                  \ifdim \@colroom>\@reqcolroom
⟨latexrelease | fltrace | flafter⟩                   \global \advance \@colnum \m@ne
⟨latexrelease | fltrace | flafter⟩                   \global \advance
⟨latexrelease | fltrace | flafter⟩                    \@textfloatsheight\ht\@currbox
```

This may sometimes give an overestimate.

```
⟨latexrelease | fltrace | flafter⟩                   \global \advance
⟨latexrelease | fltrace | flafter⟩                    \@textfloatsheight 2\intextsep
⟨latexrelease | fltrace | flafter⟩                    \@cons \@midlist \@currbox
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩                  \fl@trace{***Success: here}%
⟨latexrelease | fltrace | flafter⟩                  \fl@trace{textfloatsheight
⟨latexrelease | fltrace | flafter⟩                       (after-here) =
⟨latexrelease | fltrace | flafter⟩                       \the \@textfloatsheight}%
⟨latexrelease | fltrace | flafter⟩                  \fl@trace{colnum (after-here) =
```

```
⟨latexrelease | fltrace | flafter⟩                    \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 between paragraphs (i.e. when in vertical mode).

If the nobreak switch is true we need to reset it and clear `\everypar` since the float may not reset the flag and cannot reset the `\everypar` globally.

Typesetting starts here (we are in vertical mode).

```
⟨latexrelease | fltrace | flafter⟩                \if@nobreak
⟨latexrelease | fltrace | flafter⟩                  \nobreak
⟨latexrelease | fltrace | flafter⟩                  \@nobreakfalse
⟨latexrelease | fltrace | flafter⟩                  \everypar{}%
⟨latexrelease | fltrace | flafter⟩                \else
⟨latexrelease | fltrace | flafter⟩                  \addpenalty\interlinepenalty
⟨latexrelease | fltrace | flafter⟩                \fi
⟨latexrelease | fltrace | flafter⟩                \vskip \intextsep
⟨latexrelease | fltrace | flafter⟩                \box\@currbox
⟨latexrelease | fltrace | flafter⟩                \penalty\interlinepenalty
⟨latexrelease | fltrace | flafter⟩                \vskip\intextsep
⟨latexrelease | fltrace | flafter⟩                \ifnum\outputpenalty
⟨latexrelease | fltrace | flafter⟩                            <-\@Mii \vskip
⟨latexrelease | fltrace | flafter⟩                     -\parskip\fi
```

Typesetting ends here.

```
⟨latexrelease | fltrace | flafter⟩                \outputpenalty \z@
⟨latexrelease | fltrace | flafter⟩                \@inserttrue
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩              \else
⟨latexrelease | fltrace | flafter⟩  \fl@trace{Fail---no room at 2nd test of colroom
⟨latexrelease | fltrace | flafter⟩             (addtocorcol \string\intextsep)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩              \fi
⟨latexrelease | fltrace | flafter⟩            \fi
⟨latexrelease | fltrace | flafter⟩            \if@insert
⟨latexrelease | fltrace | flafter⟩            \else
```

Next set of docstrip guards are a bit weird, essentially `\@addtotoporbot` ends up inside the kernel and the `fltrace` package and `\@addtotoporbot` shows up in the `flafter` package. Guess that could have been done a bit more obvious :-)

```
⟨*2ekernel | fltrace⟩
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩     \fl@trace{not here: sent to addtotoporbot}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩              \@addtotoporbot
⟨/2ekernel | fltrace⟩
⟨*!2ekernel&!autoload&!fltrace⟩
```

```
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩        \fl@trace{not here: sent to addtobot}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩                  \@addtobot
⟨/!2ekernel&!autoload&!fltrace⟩
⟨latexrelease | fltrace | flafter⟩                \fi
⟨latexrelease | fltrace | flafter⟩              \fi
⟨latexrelease | fltrace | flafter⟩            \fi
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩          \else
⟨latexrelease | fltrace | flafter⟩          \fl@trace{Fail: colnum = \the \@colnum:
⟨latexrelease | fltrace | flafter⟩                     fpstype \the \@fpstype=ORD?}%
⟨latexrelease | fltrace | flafter⟩          \ifnum \@fpstype<\sixt@@n
⟨latexrelease | fltrace | flafter⟩   \fl@trace{ERROR: BANG float not successful
⟨latexrelease | fltrace | flafter⟩                                   (addtocurcol)}%
⟨latexrelease | fltrace | flafter⟩            \fi
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩          \fi
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩        \else
⟨latexrelease | fltrace | flafter⟩          \fl@trace{Fail---no room: fl box ht:
⟨latexrelease | fltrace | flafter⟩                \the \ht \@currbox (addtocurcol)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩        \fi
⟨latexrelease | fltrace | flafter⟩      \fi
⟨latexrelease | fltrace | flafter⟩    \fi
⟨latexrelease | fltrace | flafter⟩    \if@insert
⟨latexrelease | fltrace | flafter⟩    \else
⟨latexrelease | fltrace | flafter⟩      \@resethfps
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩      \fl@trace{put on deferlist (addtocurcol)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩      \@cons\@deferlist\@currbox
⟨*trace⟩
⟨latexrelease | fltrace | flafter⟩      \fl@trace{deferlist: \@deferlist:
⟨latexrelease | fltrace | flafter⟩                          (addtocurcol-after)}%
⟨/trace⟩
⟨latexrelease | fltrace | flafter⟩    \fi
⟨latexrelease | fltrace | flafter⟩   }%
⟨latexrelease | fltrace | flafter⟩\EndIncludeInRelease
```

(*End definition for* `\@addtocurcol`. *This function is documented on page* **??**.)

`\@addtonextcol` Lots of changes.

```
⟨latexrelease | fltrace⟩\IncludeInRelease{2015/01/01}
⟨latexrelease | fltrace⟩   {\@addtonextcol}{float order in 2-column}%
⟨*2ekernel | latexrelease | fltrace⟩
\def\@addtonextcol{%
  \begingroup
⟨*trace⟩
    \fl@trace{***Start addtonextcol}%
⟨/trace⟩
    \@insertfalse
    \@setfloattypecounts
```

```
   \ifnum \@fpstype=8
⟨*trace⟩
     \fl@trace{fpstype not curcol: \the \@fpstype = 8?}%
⟨/trace⟩
   \else
     \ifnum \@fpstype=24
⟨*trace⟩
       \fl@trace{fpstype not curcol: \the \@fpstype = 24?}%
⟨/trace⟩
     \else
       \@flsettextmin
⟨*trace⟩
       \fl@trace{text-so-far: Opt (top of col)}%
⟨/trace⟩
       \@reqcolroom \ht\@currbox
⟨*trace⟩
       \fl@trace{float size: \the \@reqcolroom (addtonextcol)}%
⟨/trace⟩
       \advance \@reqcolroom \@textmin
⟨*trace⟩
       \fl@trace{colroom = \the \@colroom (addtonextcol)}%
       \fl@trace{reqcolroom = \the \@reqcolroom (addtonextcol)}%
⟨/trace⟩
       \ifdim \@colroom>\@reqcolroom
         \@flsetnum \@colnum
         \ifnum\@colnum>\z@
            \@bitor\@currtype\@deferlist
⟨*trace⟩
            \fl@trace{deferlist: \@deferlist: (addtonextcol-before)}%
⟨/trace⟩
            \@testwrongwidth\@currbox
            \if@test
⟨*trace⟩
              \fl@trace{type already on list: defer (addtonextcol)}%
⟨/trace⟩
            \else
⟨*trace⟩
              \fl@trace{sent to addtotoporbot (addtonextcol)}%
⟨/trace⟩
              \@addtotoporbot
            \fi
         \fi
⟨*trace⟩
       \else
         \fl@trace{Fail---no room: fl box ht: \the \ht \@currbox
                                                 (addtonextcol)}%
⟨/trace⟩
       \fi
     \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
```

```
      \fl@trace{put back on deferlist (addtonextcol)}%
⟨/trace⟩
      \@cons\@deferlist\@currbox
⟨*trace⟩
      \fl@trace{deferlist: \@deferlist: (addtonextcol-after)}%
⟨/trace⟩
    \fi
⟨*trace⟩
  \fl@trace{End of addtonextcol -- locally counts:}%
  \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
  \endgroup
⟨*trace⟩
 \fl@trace{End of addtonextcol -- globally counts:}%
 \fl@trace{col: \the\@colnum. top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
}%
⟨/2ekernel | latexrelease | fltrace⟩
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨latexrelease | fltrace⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace⟩  {\@addtonextcol}{float order in 2-column}%
⟨latexrelease | fltrace⟩\def\@addtonextcol{%
⟨latexrelease | fltrace⟩  \begingroup
⟨*trace⟩
⟨latexrelease | fltrace⟩   \fl@trace{***Start addtonextcol}%
⟨/trace⟩
⟨latexrelease | fltrace⟩   \@insertfalse
⟨latexrelease | fltrace⟩   \@setfloattypecounts
⟨latexrelease | fltrace⟩   \ifnum \@fpstype=8
⟨*trace⟩
⟨latexrelease | fltrace⟩     \fl@trace{fpstype not curcol:
⟨latexrelease | fltrace⟩                    \the \@fpstype = 8?}%
⟨/trace⟩
⟨latexrelease | fltrace⟩   \else
⟨latexrelease | fltrace⟩     \ifnum \@fpstype=24
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{fpstype not curcol:
⟨latexrelease | fltrace⟩                        \the \@fpstype = 24?}%
⟨/trace⟩
⟨latexrelease | fltrace⟩     \else
⟨latexrelease | fltrace⟩       \@flsettextmin
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{text-so-far: 0pt (top of col)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩       \@reqcolroom \ht\@currbox
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{float size:
⟨latexrelease | fltrace⟩               \the \@reqcolroom (addtonextcol)}%
⟨latexrelease | fltrace⟩
⟨/trace⟩
⟨latexrelease | fltrace⟩       \advance \@reqcolroom \@textmin
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{colroom =
⟨latexrelease | fltrace⟩                  \the \@colroom (addtonextcol)}%
```

```
⟨latexrelease | fltrace⟩        \fl@trace{reqcolroom =
⟨latexrelease | fltrace⟩                \the \@reqcolroom (addtonextcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩        \ifdim \@colroom>\@reqcolroom
⟨latexrelease | fltrace⟩          \@flsetnum \@colnum
⟨latexrelease | fltrace⟩          \ifnum\@colnum>\z@
⟨latexrelease | fltrace⟩             \@bitor\@currtype\@deferlist
⟨*trace⟩
⟨latexrelease | fltrace⟩             \fl@trace{deferlist: \@deferlist:
⟨latexrelease | fltrace⟩                           (addtonextcol-before)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩             \if@test
⟨*trace⟩
⟨latexrelease | fltrace⟩               \fl@trace{type already on list:
⟨latexrelease | fltrace⟩                            defer (addtonextcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩             \else
⟨*trace⟩
⟨latexrelease | fltrace⟩               \fl@trace{sent to addtotoporbot
⟨latexrelease | fltrace⟩                                (addtonextcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩               \@addtotoporbot
⟨latexrelease | fltrace⟩             \fi
⟨latexrelease | fltrace⟩          \fi
⟨*trace⟩
⟨latexrelease | fltrace⟩        \else
⟨latexrelease | fltrace⟩          \fl@trace{Fail---no room: fl box ht:
⟨latexrelease | fltrace⟩               \the \ht \@currbox (addtonextcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩        \fi
⟨latexrelease | fltrace⟩      \fi
⟨latexrelease | fltrace⟩    \fi
⟨latexrelease | fltrace⟩    \if@insert
⟨latexrelease | fltrace⟩    \else
⟨*trace⟩
⟨latexrelease | fltrace⟩      \fl@trace{put back on deferlist
⟨latexrelease | fltrace⟩                                 (addtonextcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩      \@cons\@deferlist\@currbox
⟨*trace⟩
⟨latexrelease | fltrace⟩      \fl@trace{deferlist: \@deferlist:
⟨latexrelease | fltrace⟩                            (addtonextcol-after)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩    \fi
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{End of addtonextcol --
⟨latexrelease | fltrace⟩                              locally counts:}%
⟨latexrelease | fltrace⟩    \fl@trace{col: \the \@colnum.
⟨latexrelease | fltrace⟩       top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
⟨latexrelease | fltrace⟩  \endgroup
⟨*trace⟩
⟨latexrelease | fltrace⟩  \fl@trace{End of addtonextcol --
⟨latexrelease | fltrace⟩                            globally counts:}%
```

```
⟨latexrelease | fltrace⟩   \fl@trace{col: \the \@colnum.
⟨latexrelease | fltrace⟩         top: \the \@topnum. bot: \the \@botnum.}%
⟨/trace⟩
⟨latexrelease | fltrace⟩}%
⟨latexrelease | fltrace⟩\EndIncludeInRelease
```

(*End definition for* `\@addtonextcol`*. This function is documented on page* ??.)

`\@addtodblcol` Lots of changes.

```
⟨latexrelease | fltrace⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease | fltrace⟩   {\@addtodblcol}{float order in 2-column}%
⟨*2ekernel | latexrelease | fltrace⟩
\def\@addtodblcol{%
  \begingroup
⟨*trace⟩
  \fl@trace{***Start addtodblcol}%
⟨/trace⟩
   \@insertfalse
   \@setfloattypecounts
   \@getfpsbit \tw@
⟨*trace⟩
   \fl@trace{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⟩
         \fl@trace{Space OK: \@dbltoproom =
                \the \@dbltoproom > \the \ht \@currbox
                                        (dbltoproom)}%
⟨/trace⟩
       \else
⟨*trace⟩
         \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
⟨/trace⟩
         \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
           \fl@trace{BANG float ignoring \@dbltoproom}%
           \fl@trace{\@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⟩
           \fl@trace{Local value of texmin: \the\@textmin}%
           \fl@trace{\@spaces space on page = \the \@dbltoproom.
                           Ht float: \the \ht \@currbox-BANG}%
⟨/trace⟩
```

```
          \ifdim \@dbltoproom>\ht\@currbox
            \@tempswatrue
⟨*trace⟩
            \fl@trace{Space OK BANG: space on page =
                        \the \@dbltoproom > \the \ht \@currbox}%
          \else
            \fl@trace{fpstype: \the \@fpstype}%
            \fl@trace{Fail---no room dbltoproom-BANG?:}%
            \fl@trace{\@spaces space on page = \the \@dbltoproom.
                          Ht float: \the \ht \@currbox}%
⟨/trace⟩
          \fi
          \advance \@dbltoproom -\@textmin
⟨*trace⟩
        \else
          \fl@trace{fpstype: \the \@fpstype}%
          \fl@trace{Fail---no room dbltoproom-ORD?:}%
          \fl@trace{\@spaces \@dbltoproom = \the \@dbltoproom.
                          Ht float: \the \ht \@currbox}%
⟨/trace⟩
        \fi
      \fi
      \if@tempswa
          \@bitor \@currtype \@deferlist
⟨*trace⟩
          \fl@trace{(dbl)deferlist: \@deferlist: (before)}%
⟨/trace⟩
```

not in fixfloats?

```
        \@testwrongwidth\@currbox
          \if@test
⟨*trace⟩
             \fl@trace{type already on list: (dbl)defer}%
⟨/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⟩
             \fl@trace{dbltopnum (after) = \the \@dbltopnum}%
             \fl@trace{***Success: dbltop}%
⟨/trace⟩
             \@inserttrue
          \fi
      \fi
⟨*trace⟩
    \else
      \fl@trace{Fail: dbltopnum = \the \@dbltopnum: fpstype
                                                 \the \@fpstype=ORD?}%
```

```
        \ifnum \@fpstype<\sixt@@n
          \fl@trace{ERROR: !t float not successful (addtodblcol)}%
        \fi
⟨/trace⟩
      \fi
   \fi
   \if@insert
   \else
⟨*trace⟩
      \fl@trace{put on deferlist}%
⟨/trace⟩
      \@cons\@deferlist\@currbox
⟨*trace⟩
      \fl@trace{(dbl)deferlist: \@deferlist: (after)}%
⟨/trace⟩
   \fi
⟨*trace⟩
   \fl@trace{End of addtodblcol -- locally count:}%
   \fl@trace{ dbltop: \the \@dbltopnum.}%
⟨/trace⟩
  \endgroup
⟨*trace⟩
  \fl@trace{End of addtodblcol -- globally count:}%
  \fl@trace{dbltop: \the \@dbltopnum.}%
⟨/trace⟩
}%
⟨/2ekernel | latexrelease | fltrace⟩
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨latexrelease | fltrace⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace⟩  {\@addtodblcol}{float order in 2-column}%
⟨latexrelease | fltrace⟩\def\@addtodblcol{%
⟨latexrelease | fltrace⟩  \begingroup
⟨*trace⟩
⟨latexrelease | fltrace⟩  \fl@trace{***Start addtodblcol}%
⟨/trace⟩
⟨latexrelease | fltrace⟩   \@insertfalse
⟨latexrelease | fltrace⟩   \@setfloattypecounts
⟨latexrelease | fltrace⟩   \@getfpsbit \tw@
⟨*trace⟩
⟨latexrelease | fltrace⟩   \fl@trace{fpstype \ifodd \@tempcnta OK
⟨latexrelease | fltrace⟩                \else not \fi dbltop: \the \@fpstype}%
⟨/trace⟩
⟨latexrelease | fltrace⟩   \ifodd\@tempcnta
⟨latexrelease | fltrace⟩     \@flsetnum \@dbltopnum
⟨latexrelease | fltrace⟩     \ifnum \@dbltopnum>\z@
⟨latexrelease | fltrace⟩       \@tempswafalse
⟨latexrelease | fltrace⟩       \ifdim \@dbltoproom>\ht\@currbox
⟨latexrelease | fltrace⟩         \@tempswatrue
⟨*trace⟩
⟨latexrelease | fltrace⟩         \fl@trace{Space OK: \@dbltoproom =
⟨latexrelease | fltrace⟩                \the \@dbltoproom > \the \ht \@currbox
⟨latexrelease | fltrace⟩                                        (dbltoproom)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩       \else
```

```
⟨*trace⟩
⟨latexrelease | fltrace⟩      \fl@trace{fpstype: \the \@fpstype (addtodblcol)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩         \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
⟨latexrelease | fltrace⟩           \fl@trace{BANG float ignoring \@dbltoproom}%
⟨latexrelease | fltrace⟩           \fl@trace{\@spaces \@dbltoproom =
⟨latexrelease | fltrace⟩                    \the \@dbltoproom.
⟨latexrelease | fltrace⟩                    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`.

```
⟨latexrelease | fltrace⟩           \advance \@dbltoproom \@textmin
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{Local value of texmin: \the\@textmin}%
⟨latexrelease | fltrace⟩       \fl@trace{\@spaces space on page =
⟨latexrelease | fltrace⟩                   \the \@dbltoproom.
⟨latexrelease | fltrace⟩                    Ht float: \the \ht \@currbox-BANG}%
⟨/trace⟩
⟨latexrelease | fltrace⟩           \ifdim \@dbltoproom>\ht\@currbox
⟨latexrelease | fltrace⟩             \@tempswatrue
⟨*trace⟩
⟨latexrelease | fltrace⟩         \fl@trace{Space OK BANG: space on page =
⟨latexrelease | fltrace⟩                \the\@dbltoproom > \the\ht\@currbox}%
⟨latexrelease | fltrace⟩           \else
⟨latexrelease | fltrace⟩         \fl@trace{fpstype: \the \@fpstype}%
⟨latexrelease | fltrace⟩         \fl@trace{Fail---no room dbltoproom-BANG?:}%
⟨latexrelease | fltrace⟩         \fl@trace{\@spaces space on page =
⟨latexrelease | fltrace⟩                     \the \@dbltoproom.
⟨latexrelease | fltrace⟩                      Ht float: \the \ht \@currbox}%
⟨/trace⟩
⟨latexrelease | fltrace⟩           \fi
⟨latexrelease | fltrace⟩           \advance \@dbltoproom -\@textmin
⟨*trace⟩
⟨latexrelease | fltrace⟩         \else
⟨latexrelease | fltrace⟩           \fl@trace{fpstype: \the \@fpstype}%
⟨latexrelease | fltrace⟩           \fl@trace{Fail---no room dbltoproom-ORD?:}%
⟨latexrelease | fltrace⟩           \fl@trace{\@spaces \@dbltoproom =
⟨latexrelease | fltrace⟩              \the \@dbltoproom.
⟨latexrelease | fltrace⟩              Ht float: \the \ht \@currbox}%
⟨/trace⟩
⟨latexrelease | fltrace⟩         \fi
⟨latexrelease | fltrace⟩       \fi
⟨latexrelease | fltrace⟩       \if@tempswa
⟨latexrelease | fltrace⟩           \@bitor \@currtype \@dbldeferlist
⟨*trace⟩
⟨latexrelease | fltrace⟩           \fl@trace{dbldeferlist:
⟨latexrelease | fltrace⟩                     \@dbldeferlist: (before)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩           \if@test
⟨*trace⟩
⟨latexrelease | fltrace⟩             \fl@trace{type already on list: dbldefer}%
⟨/trace⟩
```

```
⟨latexrelease | fltrace⟩            \else
⟨latexrelease | fltrace⟩               \@tempdima -\ht\@currbox
⟨latexrelease | fltrace⟩               \advance\@tempdima
⟨latexrelease | fltrace⟩                 -\ifx \@dbltoplist\@empty
⟨latexrelease | fltrace⟩                       \dbltextfloatsep
⟨latexrelease | fltrace⟩                  \else \dblfloatsep \fi
⟨latexrelease | fltrace⟩               \global \advance \@dbltoproom \@tempdima
⟨latexrelease | fltrace⟩               \global \advance \@colht \@tempdima
⟨latexrelease | fltrace⟩               \global \advance \@dbltopnum \m@ne
⟨latexrelease | fltrace⟩               \@cons \@dbltoplist \@currbox
⟨*trace⟩
⟨latexrelease | fltrace⟩               \fl@trace{dbltopnum (after) =
⟨latexrelease | fltrace⟩                                 \the \@dbltopnum}%
⟨latexrelease | fltrace⟩               \fl@trace{***Success: dbltop}%
⟨/trace⟩
⟨latexrelease | fltrace⟩               \@inserttrue
⟨latexrelease | fltrace⟩            \fi
⟨latexrelease | fltrace⟩        \fi
⟨*trace⟩
⟨latexrelease | fltrace⟩      \else
⟨latexrelease | fltrace⟩        \fl@trace{Fail: dbltopnum = \the \@dbltopnum:
⟨latexrelease | fltrace⟩                           fpstype \the \@fpstype=ORD?}%
⟨latexrelease | fltrace⟩        \ifnum \@fpstype<\sixt@@n
⟨latexrelease | fltrace⟩          \fl@trace{ERROR: !t float not successful
⟨latexrelease | fltrace⟩                                      (addtodblcol)}%
⟨latexrelease | fltrace⟩        \fi
⟨/trace⟩
⟨latexrelease | fltrace⟩      \fi
⟨latexrelease | fltrace⟩    \fi
⟨latexrelease | fltrace⟩    \if@insert
⟨latexrelease | fltrace⟩    \else
⟨*trace⟩
⟨latexrelease | fltrace⟩      \fl@trace{put on dbldeferlist}%
⟨/trace⟩
⟨latexrelease | fltrace⟩      \@cons\@dbldeferlist\@currbox
⟨*trace⟩
⟨latexrelease | fltrace⟩      \fl@trace{dbldeferlist: \@dbldeferlist: (after)}%
⟨/trace⟩
⟨latexrelease | fltrace⟩    \fi
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{End of addtodblcol -- locally count:}%
⟨latexrelease | fltrace⟩    \fl@trace{ dbltop: \the \@dbltopnum.}%
⟨/trace⟩
⟨latexrelease | fltrace⟩  \endgroup
⟨*trace⟩
⟨latexrelease | fltrace⟩  \fl@trace{End of addtodblcol -- globally count:}%
⟨latexrelease | fltrace⟩  \fl@trace{dbltop: \the \@dbltopnum.}%
⟨/trace⟩
⟨latexrelease | fltrace⟩}%
⟨latexrelease | fltrace⟩\EndIncludeInRelease
```

(*End definition for* `\@addtodblcol`. *This function is documented on page* **??**.)

`\@addmarginpar`

```
⟨*2ekernel⟩
\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}}
```

(*End definition for* `\@addmarginpar`*. This function is documented on page* **??**.)

1.1.1 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.

```
\newinsert \@kludgeins
\global\dimen\@kludgeins \maxdimen
\global\count\@kludgeins 1000
```

(*End definition for* `\@kludgeins`*. This function is documented on page* ??.)

`\enlargethispage` `\enlargethispage*` The user command.

```
\gdef \enlargethispage {%
   \@ifstar
     {%
⟨*trace⟩
      \fl@trace{Enlarging page height * }%
⟨/trace⟩
      \@enlargepage{\hbox{\kern\p@}}}%
     {%
⟨*trace⟩
      \fl@trace{Enlarging page height exactly---}%
⟨/trace⟩
      \@enlargepage\@empty}%
}
```

(*End definition for* `\enlargethispage` *and* `\enlargethispage*`*. These functions are documented on page* ??.)

`\@enlargepage` This actually inserts the insert, after checking for extreme values of the change.

```
\gdef\@enlargepage#1#2{%
⟨*trace⟩
   \fl@trace{\@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⟩
      \fl@trace {Kludgeins added--pagegoal before: \the\pagegoal}%
⟨/trace⟩
      \@bsphack
        \insert\@kludgeins{#1\vskip-\@tempskipa}%
      \@esphack
```

This next bit is for tracing only:

```
⟨*trace⟩
      \ifvmode \par
        \fl@trace {Kludgeins added--pagegoal after: \the \pagegoal}%
      \fi
⟨/trace⟩
    \else
      \@latexerr{Page\space height\space already\space
                 too\space large}\@eha
    \fi
  \fi
}
⟨/2ekernel⟩
```

(*End definition for* `\@enlargepage`*. This function is documented on page* **??**.)

1.1.2 Float control

This part implements controllable floats and other changes to the float mechanism.

It provides, at the document level, the following command for inclusion in 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 supplies a reasonably exhaustive, and somewhat baroque, means of tracing some aspects of the float mechanism.

More tracing.

`\fl@trace` `\tracefloatsoff` `\tracefloats` `\fl@traceval` `\tracefloatvals` `\fl@tracemessage`

Set-up tracing for floats independent of other tracing as it produces mega-output. Default is no tracing.

```
⟨*fltrace⟩
\def \fl@tracemessage #1{{\let\@elt\@empty\typeout{LaTeX2e: #1}}}
\def \tracefloats{\let \fl@trace \fl@tracemessage}
\def \tracefloatsoff {\let \fl@trace \@gobble}
\tracefloatsoff
\def \fl@traceval #1{\fl@trace{\string #1 = \the #1}}
\IncludeInRelease{2015/01/01}{\tracefloatvals}%
                          {trace float vals}%
\def \tracefloatvals{%
```

As `\@dblfloatplacement` sets `\f@depth` it needs to be run inside a group, otherwise the float placement will test for the wrong value.[28]

```
 \begingroup
  \@dblfloatplacement
  \@floatplacement
  \fl@trace{***Float placement parameters:}%
  \fl@traceval\@colnum
  \fl@traceval\@colroom
  \fl@traceval\@topnum
  \fl@traceval\@toproom
  \fl@traceval\@botnum
  \fl@traceval\@botroom
  \fl@traceval\@fpmin
  \fl@trace{\string\textfraction = \textfraction}%
  \fl@traceval\@dbltopnum
  \fl@traceval\@dbltoproom
  \fl@trace{\string\textfraction = \textfraction}%
  \fl@trace{toplist: \@toplist}%
  \fl@trace{botlist: \@botlist}%
  \fl@trace{midlist: \@midlist}%
  \fl@trace{deferlist: \@deferlist}%
  \fl@trace{dbltoplist: \@dbltoplist}%
%FMi  \fl@trace{dbldeferlist: \@dbldeferlist}%
 \endgroup
}
\EndIncludeInRelease
\IncludeInRelease{0000/00/00}{\tracefloatvals}%
                          {trace float vals}%
\def \tracefloatvals{%
 \begingroup
  \@dblfloatplacement
  \@floatplacement
  \fl@trace{***Float placement parameters:}%
```

[28] This is a somewhat questionable design.

```
  \fl@traceval\@colnum
  \fl@traceval\@colroom
  \fl@traceval\@topnum
  \fl@traceval\@toproom
  \fl@traceval\@botnum
  \fl@traceval\@botroom
  \fl@traceval\@fpmin
  \fl@trace{\string\textfraction = \textfraction}%
  \fl@traceval\@dbltopnum
  \fl@traceval\@dbltoproom
  \fl@trace{\string\textfraction = \textfraction}%
  \fl@trace{toplist: \@toplist}%
  \fl@trace{botlist: \@botlist}%
  \fl@trace{midlist: \@midlist}%
  \fl@trace{deferlist: \@deferlist}%
  \fl@trace{dbltoplist: \@dbltoplist}%
% next line only in old releases
  \fl@trace{dbldeferlist: \@dbldeferlist}%
 \endgroup
}
\EndIncludeInRelease
```

We need to make sure that `fltrace` comes before `flafter` to make the tracing work.

```
\@ifpackageloaded{flafter}
 {\PackageWarningNoLine
      {fltrace}{Load 'fltrace' before 'flafter'\MessageBreak
                Attempting to recover by reloading 'flafter'}%
```

Hide the fact that `flafter` was already loaded and then request it anew.

```
    \expandafter\let\csname ver@flafter.sty\endcsname\relax
    \def\reserved@a#1{%
      \expandafter\let\csname\string#1+flafter+IIR\endcsname\relax}%
    \reserved@a\@addtocurcol
    \reserved@a\@addtonextcol
    \RequirePackage{flafter}}{}
⟨/fltrace⟩
```

As the code for `flafter` will contain tracing calls so that it works in conjunction with `fltrace` we need to provide a dummy definition for `\fl@trace` in that package.

```
⟨*flafter⟩
\providecommand\fl@trace[1]{}
⟨/flafter⟩
```

(*End definition for* `\fl@trace` *and others. These functions are documented on page* ??.)

`\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⟩
\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
}
```

(*End definition for* `\suppressfloats` *and* `\@flstop`*. These functions are documented on page* ??.)

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⟩
```

(*End definition for* `\@fpstype`*,* `\@reqcolroom`*, and* `\@textfloatsheight`*. These functions are documented on page* ??.)

`\@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 | fltrace⟩
\def \@fpsadddefault {%
⟨*trace⟩
   \fl@trace{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’)}%
}
```

(*End definition for* `\@fpsadddefault`*. This function is documented on page* ??.)

`\@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⟩
  \fl@trace{(mod 32) fpstype: \the \@fpstype}%
  \fl@trace{(mult of 32) currtype: \the \@currtype}%
% Tracing only: but some should be changed into real errors/warnings?
  \ifnum \@fpstype<\sixt@@n
    \ifnum \@fpstype=\z@
      \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 0?}%
    \fi
    \ifnum \@fpstype=\@ne
      \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 1?}%
    \fi
    \fl@trace{BANG float}%
  \else
    \ifnum \@fpstype=\sixt@@n
      \fl@trace{ERROR: no PLACEMENT, fpstype = \the \@fpstype = 16?}%
    \fi
    \ifnum \@fpstype=17
      \fl@trace{WARNING: only h, fpstype = \the \@fpstype = 17?}%
    \fi
    \fl@trace{ORD float}%
  \fi
⟨/trace⟩
}
⟨/2ekernel | fltrace⟩
```

(*End definition for* `\@setfloattypecounts`*. This function is documented on page* **??**.)

Macros for getting, testing and setting bits of the fps.

`\@getfpsbit` Sets `\@tempcnta` to required bit of `\count\@currbox`.

```
⟨*2ekernel⟩
\def \@getfpsbit {%
   \@boxfpsbit \@currbox
}
```

(*End definition for* `\@getfpsbit`*. This function is documented on page* **??**.)

`\@boxfpsbit` Used above.

```
\def \@boxfpsbit #1#2{%
   \@tempcnta \count#1%
   \divide \@tempcnta #2\relax
}
```

(*End definition for* `\@boxfpsbit`*. This function is documented on page* **??**.)

`\@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
}
```

(*End definition for* `\@testfp`*. This function is documented on page* **??**.)

`\@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⟩
```

(*End definition for* `\@setfpsbit`*. This function is documented on page* **??**.)

`\@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 | 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⟩
     \fl@trace{%
        `t' added to `\reserved@a h'- new Count: \the \count\@currbox}%
⟨/trace⟩
   \fi
}
```

(*End definition for* `\@resethfps`*. This function is documented on page* **??**.)

Special stuff for BANG floats.

`\@flsetnum` Ignores any zero float counter value in case BANG.

It uses a local assignment to the normally global counter: a bit naughty, perhaps?

These assignments are safe so long as the counter involved is only consulted once (i.e. only for the 'bang float') with the changed value. This is the case within `\@addtocurcol` because it is used only once within a call of the output routine (which forms a group).

For `\@addtonextcol` this is achieved by putting a group around its code; this is needed because it is called (by `\@startcolumn`) for each float which was on the deferlist.

Almost identical considerations pertain to \@addtodblcol. There may be more efficient ways to handle this, but the group seems to be the simplest.

```
\def \@flsetnum #1{%
⟨*trace⟩
   \fl@trace{fpstype: \the \@fpstype (flsetnum \string#1)}%
⟨/trace⟩
   \ifnum \@fpstype<\sixt@@n
     \ifnum #1=\z@
⟨*trace⟩
       \fl@trace{BANG float resetting \string#1 to 1}%
⟨/trace⟩
       #1\@ne
     \fi
   \fi
⟨*trace⟩
   \fl@trace{#1 (before) = \the #1}%
⟨/trace⟩
}
```

(*End definition for* \@flsetnum. *This function is documented on page* ??.)

\@flsettextmin This ignores \textfraction space restriction in case BANG.

```
\def \@flsettextmin {%
⟨*trace⟩
   \fl@trace{fpstype: \the \@fpstype (flsettextmin)}%
⟨/trace⟩
   \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
     \fl@trace{BANG ignoring textmin}%
⟨/trace⟩
     \@textmin \z@
   \else
     \@textmin \textfraction\@colht
⟨*trace⟩
     \fl@trace{ORD textmin = \the \@textmin}%
⟨/trace⟩
   \fi
}
```

(*End definition for* \@flsettextmin. *This function is documented on page* ??.)

\@flcheckspace This ignores space restriction in case BANG; this is still slightly conservative since it does not allow for the fact that, if there is no text in the column then \textfloatsep is not needed. Sets @tempswa true if there is room for \@currbox.

```
\def \@flcheckspace #1#2{%
   \advance \@reqcolroom
     \ifx #2\@empty \textfloatsep \else \floatsep \fi
⟨*trace⟩
   \fl@trace{colroom = \the \@colroom
                                  (flcheckspace \string#1 \string#2)}%
   \fl@trace{reqcolroom = \the \@reqcolroom
                                  (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
```

```
   \ifdim \@colroom>\@reqcolroom
     \ifdim #1>\ht\@currbox
       \@tempswatrue
⟨*trace⟩
       \fl@trace{Space OK: #1 = \the #1 > \the \ht \@currbox
                                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
     \else
⟨*trace⟩
       \fl@trace{fpstype: \the \@fpstype
                                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
       \ifnum \@fpstype<\sixt@@n
⟨*trace⟩
         \fl@trace{BANG float ignoring #1
                                   (flcheckspace \string#1 \string#2):}%
         \fl@trace{\@spaces #1 = \the #1.  Ht float: \the \ht \@currbox
                                                          BANG}%
⟨/trace⟩
         \@tempswatrue
⟨*trace⟩
       \else
         \fl@trace{Fail---no room (flcheckspace \string#1 \string#2)
                       (fpstype \the \@fpstype=ORD?):}%
         \fl@trace{\@spaces #1 = \the #1.  Ht float: \the \ht \@currbox
                                                          ORD?}%
⟨/trace⟩
       \fi
     \fi
⟨*trace⟩
   \else
     \fl@trace{Fail---no room at 2nd test of colroom
                   (flcheckspace \string#1 \string#2)}%
⟨/trace⟩
   \fi
}
⟨/2ekernel | fltrace⟩
```

(*End definition for* `\@flcheckspace`*. This function is documented on page* ??.)

`\@flupdates` This updates everything when a float is placed.

```
⟨*2ekernel⟩
\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⟩
```

(*End definition for* `\@flupdates`. *This function is documented on page* ??.)

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 immediately follows other spaces. This is also possibly too big a change, at least for compatibility mode? Or it may be simply wrong! It has not been changed.
9. Now `\@addtocurcol` checks first for just p fps. I think that this is an increase in efficiency, but maybe the coding should be made even more efficient.
10. `\@tryfcolumn` now tests if the list is empty first, otherwise lots of wasted time! Thus this test has been removed from `\@startcolumn`. As Frank pointed out, this makes `\@startcolumn` less efficient. But it is now the same as `\@startdblcolumn`: I can see no reason why they should be different, but which is best?
11. Why is `\@colroom` set in `\@doclearpage`?
12. Footnotes. Check what `\clearpage` does when footnotes are left over. Footnotes are not put on float pages and, also, `\@addtonextcol` ignores the existence of held-over footnotes in deciding what floats can go on the page. Not changed.
13. `\clearpage` can still lose non-boxes, at least when floats are involved. It also moves some to the 'wrong page', but this may be a coding problem.
14. The ! option makes it necessary to check in `\output` that there is enough room left on the page after adding a float. (This would have been necessary anyway if anyone set `\@textmin` too close to zero! A similar danger existed also if the text in a `\twocolumn[text]` entity gets too large.) The current implementation of this also makes the normal case a little less efficient, OK? Not enough room means, at present, less than `\baselineskip`, with a warning: is this OK? Should it be made generic (another parameter)?

15. There are four possibilities for supporting this:

    `\twocolumn[\maketitle more text]`

    One is to change `\maketitle` slightly to allow this. Another is to change `\@topnewpage` so that more than one `\twocolumn[]` command is allowed; in this case `\maketitle\twocolumn[more text]` will work. The former is more robust from the user's viewpoint, but makes the code for `\maketitle` rather ad hoc (maybe it is already?). Another is to misuse the global twocolumn flag locally within `\@topnewpage`. Yet another is to move the column count register from the multicol package into the kernel. This has been done.

16. Where should the reinserts be put to maximise the probability that 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 adjustment for the depth is needed if boxmaxdepth is used to ensure that there is never a too deep box.

21. The value of `\boxmaxdepth` should be explicitly set whenever necessary: it is too risky to assume that it has any particular value. Care is needed in deciding what to set it to.

    It is interesting to note that the value of `\boxmaxdepth` is unique in being read before the local settings for the box group are reset; all other parameter settings which affect the box construction use their values outside the box group.

22. Should `\@maxdepth` store the setting of `\maxdepth` from lplain? Or should we provide a proper interface to class files for setting these?

An analysis of various other macros.

`\@opcol` should do `\@floatplacement`, but where? Right at the end, since it always occurs at the start of a column.

```
\def\@opcol{%
  % Why is this done first?
  \global \@mparbottom \z@
  \if@twocolumn
```

```
    \@outputdblcol
  \else
    \@outputpage
    % This is not needed since it is done at the end of
    %   |\@outputpage|:
    \global \@colht \textheight
  \fi}
```

Only tracing has been added to these.

```
⟨latexrelease | fltrace⟩\IncludeInRelease{2017/01/01}%
⟨latexrelease | fltrace⟩  {\@makefcolumn}{negative height floats}%
⟨*2ekernel | fltrace | latexrelease⟩
\def\@makefcolumn #1{%
  \begingroup
    \@fpmin -\maxdimen
    \let \@testfp \@gobble
    \@tryfcolumn #1%
  \endgroup
⟨*trace⟩
  \if@fcolmade
    \fl@trace{PAGE: in \string\clearpage
                              \if@twocolumn ---twocolumn\fi---}%
    \fl@trace{----- float column/page completed from \string#1}%
  \fi
⟨/trace⟩
}
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨latexrelease | fltrace⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace⟩  {\@makefcolumn}{negative height floats}%
⟨latexrelease | fltrace⟩\def\@makefcolumn #1{%
⟨latexrelease | fltrace⟩  \begingroup
⟨latexrelease | fltrace⟩    \@fpmin \z@
⟨latexrelease | fltrace⟩    \let \@testfp \@gobble
⟨latexrelease | fltrace⟩    \@tryfcolumn #1%
⟨latexrelease | fltrace⟩  \endgroup
⟨*trace⟩
⟨latexrelease | fltrace⟩  \if@fcolmade
⟨latexrelease | fltrace⟩    \fl@trace{PAGE: in \string\clearpage
⟨latexrelease | fltrace⟩                      \if@twocolumn ---twocolumn\fi---}%
⟨latexrelease | fltrace⟩    \fl@trace{----- float column/page completed
⟨latexrelease | fltrace⟩                      from \string#1}%
⟨latexrelease | fltrace⟩  \fi
⟨/trace⟩
⟨latexrelease | fltrace⟩}
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨/2ekernel | fltrace | latexrelease⟩
```

This will line up the last baselines in the two columns provided they are constructed in the normal way: i.e. ending in a skip of minus the original depth, with `\@textbottom` adding nothing.

Thus again it is essential for `\@textbottom` to have depth 0pt.

```
⟨latexrelease | fltrace⟩\IncludeInRelease{2015/01/01}%
```

```
⟨latexrelease | fltrace⟩   {\@outputdblcol}{2 column marks}%
⟨*2ekernel | fltrace | latexrelease⟩
```

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```
\def\@outputdblcol{%
  \if@firstcolumn
    \global\@firstcolumnfalse
```

Save the left column

```
    \global\setbox\@leftcolumn\copy\@outputbox
⟨fltrace⟩     \fl@trace{PAGE: first column boxed}%
```

Remember the marks from the first column

```
    \splitmaxdepth\maxdimen
    \vbadness\maxdimen
```

In case of `\enlargethispage` we will have infinite negative glue at the bottom of the page (coming from `\vss`) and that will earn us an error message if we `\vsplit` to get at the marks. So we need to remove the last glue (if any) at the end of `\@outputbox` as we are only interested in marks that change doesn't matter.

```
      \setbox\@outputbox\vbox{\unvbox\@outputbox\unskip}%
      \setbox\@outputbox\vsplit\@outputbox to\maxdimen
```

One minor difference from the current `fixmarks` package, pass the marks through a token register to stop any `#` tokens causing an error in a `\def`.

```
    \toks@\expandafter{\topmark}%
    \xdef\@firstcoltopmark{\the\toks@}%
    \toks@\expandafter{\splitfirstmark}%
    \xdef\@firstcolfirstmark{\the\toks@}%
```

This test does not work if truly empty marks have been inserted, but LaTeX marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```
    \ifx\@firstcolfirstmark\@empty
      \global\let\@setmarks\relax
    \else
      \gdef\@setmarks{%
        \let\firstmark\@firstcolfirstmark
        \let\topmark\@firstcoltopmark}%
    \fi
```

End of change

```
  \else
    \global\@firstcolumntrue
    \setbox\@outputbox\vbox{%
     \hb@xt@\textwidth{%
        \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
        \hfil
```

The color of the `\vrule` should be `\normalcolor` as to not inherit the color from the column.

```
        {\normalcolor\vrule \@width\columnseprule}%
        \hfil
       \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
⟨fltrace⟩     \fl@trace{PAGE: second column also boxed}%
  \@combinedblfloats
```

Override current first and top with those of first column if necessary

```
    \@setmarks
```

End of change

```
    \@outputpage
⟨fltrace⟩    \fl@trace{PAGE: two column page completed}%
    \begingroup
      \@dblfloatplacement
      \@startdblcolumn
      \@whilesw\if@fcolmade \fi{\@outputpage
⟨fltrace⟩      \fl@trace{PAGE: double float page completed}%
     \@startdblcolumn}%
    \endgroup
  \fi}%
```

```
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨latexrelease | fltrace⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease | fltrace⟩  {\@outputdblcol}{2 column marks}%
⟨latexrelease | fltrace⟩\def\@outputdblcol{%
⟨latexrelease | fltrace⟩  \if@firstcolumn
⟨latexrelease | fltrace⟩    \global \@firstcolumnfalse
⟨latexrelease | fltrace⟩    \global \setbox\@leftcolumn \box\@outputbox
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{PAGE: first column boxed}%
⟨/trace⟩
⟨latexrelease | fltrace⟩  \else
⟨latexrelease | fltrace⟩    \global \@firstcolumntrue
⟨latexrelease | fltrace⟩    \setbox\@outputbox \vbox {%
⟨latexrelease | fltrace⟩                        \hb@xt@\textwidth {%
⟨latexrelease | fltrace⟩                          \hb@xt@\columnwidth {%
⟨latexrelease | fltrace⟩                            \box\@leftcolumn \hss}%
⟨latexrelease | fltrace⟩                          \hfil
⟨latexrelease | fltrace⟩                          {\normalcolor\vrule
⟨latexrelease | fltrace⟩                              \@width\columnseprule}%
⟨latexrelease | fltrace⟩                          \hfil
⟨latexrelease | fltrace⟩                          \hb@xt@\columnwidth {%
⟨latexrelease | fltrace⟩                            \box\@outputbox \hss}%
⟨latexrelease | fltrace⟩                                                }%
⟨latexrelease | fltrace⟩                             }%
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{PAGE: second column also boxed}%
⟨/trace⟩
⟨latexrelease | fltrace⟩    \@combinedblfloats
⟨latexrelease | fltrace⟩    \@outputpage
⟨*trace⟩
⟨latexrelease | fltrace⟩    \fl@trace{PAGE: two column page completed}%
⟨/trace⟩
⟨latexrelease | fltrace⟩    \begingroup
⟨latexrelease | fltrace⟩      \@dblfloatplacement
⟨latexrelease | fltrace⟩      \@startdblcolumn
```

This loop could be replaced by an `\expandafter` tail recursion in `\@startdblcolumn`.

```
⟨latexrelease | fltrace⟩      \@whilesw\if@fcolmade \fi
⟨latexrelease | fltrace⟩        {\@outputpage
```

```
⟨*trace⟩
⟨latexrelease | fltrace⟩       \fl@trace{PAGE: double float page completed}%
⟨/trace⟩
⟨latexrelease | fltrace⟩          \@startdblcolumn}%
⟨latexrelease | fltrace⟩     \endgroup
⟨latexrelease | fltrace⟩   \fi
⟨latexrelease | fltrace⟩}%
⟨latexrelease | fltrace⟩\EndIncludeInRelease
⟨/2ekernel | fltrace | latexrelease⟩
```

1.1.3 Float placement parameters

The main purpose of this section is to ensure that all the float-placement parameters which need to be set in a class file or package have been declared. It also describes their use and sets values for them which are reasonable for typical documents using US letter or A4 sized paper.

Limits for the placement of floating objects

`\c@topnumber` This counter holds the maximum number of floats that can appear at the top of a text page or column.

```
⟨*2ekernel⟩
\newcount\c@topnumber
\setcounter{topnumber}{2}
```

(*End definition for* `\c@topnumber`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\topfraction`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\c@bottomnumber`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\bottomfraction`*. This function is documented on page* **??**.)

`\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}
```

(*End definition for* `\c@totalnumber`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\textfraction`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\floatpagefraction`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\c@dbltopnumber`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\dbltopfraction`*. This function is documented on page* **??**.)

\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}
```

(*End definition for* `\dblfloatpagefraction`*. This function is documented on page* **??**.)

Floats on a text page

\floatsep
\textfloatsep
\intextsep

When a floating object is placed on a page with text, these parameters control the separation between the float and the other objects on the page. These parameters are used for both one-column mode and single-column floats in two-column mode. They are all rubber lengths.

`\floatsep` is the space between adjacent floats that are placed at the top or bottom of the text page or column.

`\textfloatsep` is the space between the main text and floats at the top or bottom of the page or column.

`\intextsep` is the space between in-text floats and the text.

```
\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@}
```

(*End definition for* `\floatsep`*,* `\textfloatsep`*, and* `\intextsep`*. These functions are documented on page* **??**.)

\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@}
```

(*End definition for* \dblfloatsep *and* \dbltextfloatsep*. These functions are documented on page* ??.)

Floats on their own page or column

\@fptop
\@fpsep
\@fpbot

When floating objects are placed on a separate page or column, called a 'float page', the layout of the page is controlled by these parameters, which are rubber lengths.

At the top of the page \@fptop is inserted; typically this supplies some stretchable whitespace. At the bottom of the page \@fpbot 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}
```

(*End definition for* \@fptop*,* \@fpsep*, and* \@fpbot*. These functions are documented on page* ??.)

\@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}
```

(*End definition for* \@dblfptop*,* \@dblfpsep*, and* \@dblfpbot*. These functions are documented on page* ??.)

\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⟩
```

(*End definition for* \topfigrule*,* \botfigrule*, and* \dblfigrule*. These functions are documented on page* ??.)

File T
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 2_ε.

```
⟨*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 2_ε 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 U
ltfinal.dtx

1 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.

1.1 Debugging

By default, LaTeX shows statistics:

```
⟨*2ekernel⟩
\tracingstats1
```

1.2 Typesetting parameters

`\@lowpenalty` `\@medpenalty` `\@highpenalty`

These are penalties used internally.

```
\newcount\@lowpenalty
\newcount\@medpenalty
\newcount\@highpenalty
```

(*End definition for* `\@lowpenalty`*,* `\@medpenalty`*, and* `\@highpenalty`*. These functions are documented on page* **??**.)

`\newmarks`

Allocate extended marks types if etex is active. Placed here at the end of the format to increase compatibility with count allocations in earlier releases.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                  {\newmarks}{Extended Allocation}%
\ifx\marks\@undefined\else
\def\newmarks{%
  \e@alloc\marks \e@alloc@chardef{\count256}\m@ne\e@alloc@top}
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\newmarks}{Extended Allocation}%
⟨latexrelease⟩\let\newmarks\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\newmarks`*. This function is documented on page* **??**.)

`\newXeTeXintercharclass` `\xe@alloc@intercharclass` `\e@alloc@intercharclass@top`

Allocate `\XeTeXintercharclass` types if xetex is active. previously defined in `xetex.ini`.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩                {\newXeTeXintercharclass}{Extended Allocation}%
```

Classes allocated 1 to 4094 (or 254 on older xetex) (In earlier XeLaTeX versions 1, 2 and 3 were pre-set for CJK).

```
\ifx\XeTeXcharclass\@undefined
\else
\ifdim\the\XeTeXversion\XeTeXrevision\p@>0.99993\p@
  \chardef\e@alloc@intercharclass@top=4095
\else
  \chardef\e@alloc@intercharclass@top=255
\fi
\def\newXeTeXintercharclass{%
 \e@alloc\XeTeXcharclass
   \chardef\xe@alloc@intercharclass\m@ne\e@alloc@intercharclass@top}
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩              {\newXeTeXintercharclass}{Extended Allocation}%
⟨latexrelease⟩ \ifx\XeTeXcharclass\@undefined
⟨latexrelease⟩ \else
⟨latexrelease⟩    \def\xe@alloc@#1#2#3#4#5{\global\advance#1\@ne
⟨latexrelease⟩     \xe@ch@ck#1#4#2%
⟨latexrelease⟩     \allocationnumber#1%
⟨latexrelease⟩     \global#3#5\allocationnumber
⟨latexrelease⟩     \wlog{\string#5=\string#2\the\allocationnumber}}
⟨latexrelease⟩    \def\xe@ch@ck#1#2#3{%
⟨latexrelease⟩     \ifnum#1<#2\else
⟨latexrelease⟩      \errmessage{No room for a new #3}%
⟨latexrelease⟩     \fi}
⟨latexrelease⟩    \def\newXeTeXintercharclass{%
⟨latexrelease⟩     \xe@alloc@\xe@alloc@intercharclass
⟨latexrelease⟩                    \XeTeXcharclass\chardef\@cclv}
⟨latexrelease⟩ \fi
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2016/02/01}%
⟨latexrelease⟩  {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
\ifx\XeTeXcharclass\@undefined
\else
  \countdef\xe@alloc@intercharclass=257
  \xe@alloc@intercharclass=\z@
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{2015/01/01}%
⟨latexrelease⟩  {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
⟨latexrelease⟩ \ifx\XeTeXcharclass\@undefined
⟨latexrelease⟩ \else
⟨latexrelease⟩   \xe@alloc@intercharclass=\thr@@
⟨latexrelease⟩ \fi
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\xe@alloc@intercharclass}{Start of XeTeX class allocator}%
⟨latexrelease⟩ \ifx\XeTeXcharclass\@undefined
```

```
⟨latexrelease⟩ \else
⟨latexrelease⟩   \newcount\xe@alloc@intercharclass
⟨latexrelease⟩   \xe@alloc@intercharclass=\thr@@
⟨latexrelease⟩ \fi
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\newXeTeXintercharclass`, `\xe@alloc@intercharclass`, *and* `\e@alloc@intercharclass@top`. *These functions are documented on page* **??**.)

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
```

1.3 Lccodes for hyphenation

For 7- and 8-bit engines the assumption of T1 encodings is the basis for the hyphenation patterns. That's not the case for the Unicode engines, where the assumption is engine-native working. The common loader system provides access to data from the Unicode Consortium covering not only `\lccode` but also other related data. The `\lccode` part of that at least needs to be loaded before hyphenation is tackled: XeTEX follows the standard TEX route of building patterns into the format. LuaTEX doesn't require this data be loaded *here* but it does need to be loaded somewhere. Rather than test for the Unicode engines by name, the approach here is to look for the extended math mode handling both provide: any other engine developed in this area will presumably also provide `\Umathcode`.

```
\ifnum 0%
  \ifx\Umathcode\@undefined\else 1\fi
  \ifx\XeTeXmathcode\@undefined\else 1\fi
  >\z@
  \message{ Unicode character data,}
  \input{load-unicode-data}
⟨/2ekernel⟩
⟨latexrelease⟩\IncludeInRelease{2016/02/01}%
⟨latexrelease⟩  {\XeTeXintercharclasses}{XeTeX character classes}%
⟨latexrelease⟩  \ifx\XeTeXinterchartoks\undefined
⟨latexrelease⟩  \else
```

```
⟨latexrelease⟩    \begingroup
⟨latexrelease⟩      \chardef\XeTeXcharclassID = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassOP = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassCL = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassEX = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassIS = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassNS = 0 %
⟨latexrelease⟩      \chardef\XeTeXcharclassCM = 0 %
⟨latexrelease⟩      \input{load-unicode-xetex-classes}
⟨latexrelease⟩    \endgroup
⟨latexrelease⟩    \global\let\xtxHanGlue\undefined
⟨latexrelease⟩    \global\let\xtxHanSpace\undefined
⟨latexrelease⟩    \global\XeTeXinterchartoks 0 1 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 0 2 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 0 3 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 1 0 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 2 0 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 3 0 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 1 1 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 1 2 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 1 3 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 2 1 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 2 2 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 2 3 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 3 1 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 3 2 = {}
⟨latexrelease⟩    \global\XeTeXinterchartoks 3 3 = {}
⟨latexrelease⟩  \fi
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩  {\XeTeXintercharclasses}{XeTeX character classes}%
⟨latexrelease⟩  \ifx\XeTeXinterchartoks\undefined
⟨latexrelease⟩  \else
⟨latexrelease⟩   \input{load-unicode-xetex-classes}
⟨latexrelease⟩   \gdef\xtxHanGlue{\hskip0pt plus 0.1em\relax}
⟨latexrelease⟩   \gdef\xtxHanSpace{\hskip0.2em plus 0.2em minus 0.1em\relax}
⟨latexrelease⟩   \global\XeTeXinterchartoks 0 1 = {\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 0 2 = {\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 0 3 = {\nobreak\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 1 0 = {\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 2 0 = {\nobreak\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 3 0 = {\xtxHanSpace}
⟨latexrelease⟩   \global\XeTeXinterchartoks 1 1 = {\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 1 2 = {\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 1 3 = {\nobreak\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 2 1 = {\nobreak\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 2 2 = {\nobreak\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 2 3 = {\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 3 1 = {\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 3 2 = {\xtxHanGlue}
⟨latexrelease⟩   \global\XeTeXinterchartoks 3 3 = {\nobreak\xtxHanGlue}
⟨latexrelease⟩  \fi
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

There is one over-ride that makes sense here (see below for the same for 8-bit engines): setting the lccode for - to itself.

```
  \lccode`\- =`\- % default hyphen char
```

The alternative is that a "traditional" engine is in use.

```
\else
```

We set things up so that hyphenation files can assume that the default (T1) lccodes are in use (at present this also sets up the uccodes). We temporarily define `\reserved@a` to apply `\reserved@c` to all the numbers in the range of its arguments.

```
\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}
\reserved@a{"A0}{"BC}
\reserved@a{"E0}{"FF}
```

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}
\reserved@a{"80}{"9C}
\reserved@a{"C0}{"DF}
```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose uccode or lccode isn't quite what you'd expect.

```
\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
\lccode`\^^9d=`\i    % dotted I
\uccode`\^^9d=`\^^9d % dotted I
\lccode`\^^9e=`\^^9e % d-bar
\uccode`\^^9e=`\^^d0 % d-bar
```

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
```

End of the conditional to select either Unicode or T1 encoding defaults.

```
\fi
```

At this stage, we can install any last-minute expl3 set-up.

```
\@expl@finalise@setup@@
\def\@expl@finalise@setup@@{}
```

This is as good a place as any to active a few XeTEX-specific settings

```
\ifx\XeTeXuseglyphmetrics\@undefined
\else
  \XeTeXuseglyphmetrics=1 %
  \XeTeXdashbreakstate=1 %
\fi
```

1.4 Hyphenation

The following code will be compiled into the format file. It checks for the existence of `hyphen.cfg` in inputs that file if found. Otherwise it inputs `hyphen.ltx`. Note that these are loaded in *before* the `\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
```

`\l@nohyphenation`

```
\ifx\l@nohyphenation \@undefined
  \newlanguage\l@nohyphenation
\fi
```

(*End definition for* `\l@nohyphenation`*. This function is documented on page* **??**.)

`\document@default@language` Default document language. -1 acts as language 0, but used as a flag in `\document` to see if it has been set in the preamble.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
⟨latexrelease⟩\IncludeInRelease{2017/04/15}%
⟨latexrelease⟩                  {\document@default@language}{Save language for hyphenation}%
\let\document@default@language\m@ne
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\document@default@language}{Save language for hyphenation}%
⟨latexrelease⟩\let\document@default@language\@undefined
⟨latexrelease⟩\EndIncludeInRelease
⟨*2ekernel⟩
```

(*End definition for* `\document@default@language`*. This function is documented on page* ??.)

1.5 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.

```
\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
```

1.6 Input encoding

Starting with the 2018 LaTeX release default the inputencoding to UTF-8. Unless the format is being used with luatex, xetex, enctex or mltex.

This is done in a way largely compatible with older releases: `utf8.def` is input just as if
`\usepackage[utf8]{inputenc}`
had been used, however rather than input the whole package a minimal core part just enough to support loading the UTF-8 encoding files is defined here.

If a document re-specifies UTF-8 this is silently ignored.

```
⟨/2ekernel⟩
⟨*2ekernel | latexrelease⟩
```

Check that a classic 8-bit tex engine is being used (LaTeX or PDFLaTeX).

```
⟨latexrelease⟩\IncludeInRelease{2018/04/01}%
⟨latexrelease⟩                  {\UTFviii@invalid}{UTF-8 default}%
```

Skip this section in Unicode TeX, or if MLTeX and EncTeX are enabled.

```
\ifnum0%
  \ifx\Umathcode\@undefined\else 1\fi
  \ifx\mubyte\@undefined\else 1\fi
  \ifx\charsubdef\@undefined\else 1\fi
  =\z@
```

```
\def\saved@space@catcode{10}
\let\@inpenc@test\relax
\def\IeC{%
  \ifx\protect\@typeset@protect
    \expandafter\@firstofone
  \else
    \noexpand\IeC
  \fi
}
```

Make characters active for UTF-8 input formats

```
\@tempcnta=1
\loop
  \catcode\@tempcnta=13  %
  \advance\@tempcnta\@ne %
\ifnum\@tempcnta<32      %
\repeat                  %
\catcode0=15  % null
\catcode9=10  % tab
\catcode10=12 % ctrl J
\catcode12=13 % ctrl L
\catcode13=5  % newline
\@tempcnta=128
\loop
  \catcode\@tempcnta=13
  \advance\@tempcnta\@ne
\ifnum\@tempcnta<256
\repeat
```

`\UseRawInputEncoding` Reset 8 bit characters to catcode 12 so the input endcoing matches the "Raw" font encoding. Useful for special behaviours, or for compatibility with older LaTeX formats.

```
\def\UseRawInputEncoding{%
\let\inputencodingname\@undefined                    % revert
\let\DeclareFontEncoding@\DeclareFontEncoding@saved   % revert
\let\DeclareUnicodeCharacter\@undefined               % revert
\@tempcnta=1
\loop
  \catcode\@tempcnta=15  %
  \advance\@tempcnta\@ne %
\ifnum\@tempcnta<32      %
\repeat                  %
\catcode0=15  % null
\catcode9=10  % tab
\catcode10=12 % ctrl J
\catcode12=13 % ctrl L
\catcode13=5  % newline
\@tempcnta=128
\loop
  \catcode\@tempcnta=12
  \advance\@tempcnta\@ne
\ifnum\@tempcnta<256
\repeat
}
```

(*End definition for* `\UseRawInputEncoding`*. This function is documented on page* **??**.)

\DeclareFontEncoding@saved Saved version of \DeclareFontEncoding@ before utf8.def modifies it for use in \UseRawInputEncoding above.

```
\let\DeclareFontEncoding@saved\DeclareFontEncoding@
```

(*End definition for* \DeclareFontEncoding@saved. *This function is documented on page* ??.)

```
\edef\inputencodingname{utf8}%
\input{utf8.def}
\let\UTFviii@undefined@err@@\UTFviii@undefined@err
\let\UTFviii@invalid@err@@\UTFviii@invalid@err
\let\UTFviii@two@octets@@\UTFviii@two@octets
\let\UTFviii@three@octets@@\UTFviii@three@octets
\let\UTFviii@four@octets@@\UTFviii@four@octets
⟨2ekernel⟩\def\UTFviii@undefined@err#1{\@gobble#1}%
⟨2ekernel⟩\let\UTFviii@invalid@err\string
⟨2ekernel⟩\let\UTFviii@two@octets\string
⟨2ekernel⟩\let\UTFviii@three@octets\string
⟨2ekernel⟩\let\UTFviii@four@octets\string
⟨2ekernel⟩\everyjob\expandafter{\the\everyjob
⟨2ekernel⟩\let\UTFviii@undefined@err\UTFviii@undefined@err@@
⟨2ekernel⟩\let\UTFviii@invalid@err\UTFviii@invalid@err@@
⟨2ekernel⟩\let\UTFviii@two@octets\UTFviii@two@octets@@
⟨2ekernel⟩\let\UTFviii@three@octets\UTFviii@three@octets@@
⟨2ekernel⟩\let\UTFviii@four@octets\UTFviii@four@octets@@
⟨2ekernel⟩}
\let\@inpenc@test\@undefined
\let\saved@space@catcode\@undefined
```

For formats not set up for UTF-8 default, set the C0 controls to catcode 15.

```
\else
\@tempcnta=0
\loop
  \catcode\@tempcnta=15  %
  \advance\@tempcnta\@ne %
\ifnum\@tempcnta<32      %
\repeat                  %
\catcode0=15  % null
\catcode9=10  % tab
\catcode10=12 % ctrl J
\catcode12=13 % ctrl L
\catcode13=5  % newline
```

```
\let\UseRawInputEncoding\relax
```

This ends the skipped code in Unicode engines:

```
\fi
⟨/2ekernel | latexrelease⟩
⟨latexrelease⟩\EndIncludeInRelease
⟨latexrelease⟩\IncludeInRelease{0000/00/00}%
⟨latexrelease⟩                  {\UTFviii@invalid}{UTF-8 default}%
```

The first block of commands got only introduced in 2019 but we revert all of Unicode support in one go not jump to the intermediate version.

```
⟨latexrelease⟩  \let\UTFviii@two@octets@combine\@undefined
⟨latexrelease⟩  \let\UTFviii@three@octets@combine\@undefined
⟨latexrelease⟩  \let\UTFviii@four@octets@combine\@undefined
```

```
⟨latexrelease⟩  \let\UTFviii@two@octets@string\@undefined
⟨latexrelease⟩  \let\UTFviii@three@octets@string\@undefined
⟨latexrelease⟩  \let\UTFviii@four@octets@string\@undefined
⟨latexrelease⟩  \let\UTFviii@two@octets@noexpand\@undefined
⟨latexrelease⟩  \let\UTFviii@three@octets@noexpand\@undefined
⟨latexrelease⟩  \let\UTFviii@four@octets@noexpand\@undefined
```

```
⟨latexrelease⟩\@tempcnta=0
⟨latexrelease⟩\loop
⟨latexrelease⟩  \catcode\@tempcnta=15
⟨latexrelease⟩  \advance\@tempcnta\@ne
⟨latexrelease⟩\ifnum\@tempcnta<32
⟨latexrelease⟩\repeat       %
⟨latexrelease⟩\catcode9=10  % tab
⟨latexrelease⟩\catcode10=12 % ctrl J
⟨latexrelease⟩\catcode12=13 % ctrl L
⟨latexrelease⟩\catcode13=5  % newline
⟨latexrelease⟩\@tempcnta=128
⟨latexrelease⟩\loop
⟨latexrelease⟩\catcode\@tempcnta=12
⟨latexrelease⟩\advance\@tempcnta\@ne
⟨latexrelease⟩\ifnum\@tempcnta<256
⟨latexrelease⟩\repeat
⟨latexrelease⟩\let\IeC\@undefined
⟨latexrelease⟩\def\DeclareFontEncoding@#1#2#3{%
⟨latexrelease⟩  \expandafter
⟨latexrelease⟩  \ifx\csname T@#1\endcsname\relax
⟨latexrelease⟩     \def\cdp@elt{\noexpand\cdp@elt}%
⟨latexrelease⟩     \xdef\cdp@list{\cdp@list\cdp@elt{#1}%
⟨latexrelease⟩                    {\default@family}{\default@series}%
⟨latexrelease⟩                    {\default@shape}}%
⟨latexrelease⟩     \expandafter\let\csname#1-cmd\endcsname\@changed@cmd
⟨latexrelease⟩  \else
⟨latexrelease⟩     \@font@info{Redeclaring font encoding #1}%
⟨latexrelease⟩  \fi
⟨latexrelease⟩  \global\@namedef{T@#1}{#2}%
⟨latexrelease⟩  \global\@namedef{M@#1}{\default@M#3}%
⟨latexrelease⟩  \xdef\LastDeclaredEncoding{#1}%
⟨latexrelease⟩  }
⟨latexrelease⟩  \let\UseRawInputEncoding\@undefined
⟨latexrelease⟩  \let\DeclareFontEncoding@saved\@undefined
⟨latexrelease⟩  \let\inputencodingname\@undefined
⟨latexrelease⟩\EndIncludeInRelease
```

```
⟨*2ekernel⟩
%    \begin{macrocode}
%
% We temporarily define |\reserved@a| to apply |\reserved@c| to all the
% numbers in the range of its arguments.
%    \begin{macrocode}
\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`).

1.7 Lccodes and uccodes

We now again set up the default (T1) uc/lccodes. The lower case characters need their `\uccode` and `\lccode` values set. Some of this is a repeat of the set-up before loading hyphenation files. Depending on the TEX version, we might not be allowed to do this for non-ASCII characters. For the Unicode engines (XeTEX and LuaTEX) there is no need to do any of this: they use hyphenation data which does not alter any of the set up and so this entire block is skipped.

```
\ifnum 0%
  \ifx\Umathcode\@undefined\else 1\fi
  \ifx\XeTeXmathcode\@undefined\else 1\fi
  >\z@
\else
\def\reserved@c#1{%
   \count@=#1\advance\count@ by -"20
   \uccode#1=\count@
   \lccode#1=#1
}
\reserved@a{`\a}{`\z}
\reserved@a{"A0}{"BC}
\reserved@a{"E0}{"FF}
```

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}
\reserved@a{"80}{"9C}
\reserved@a{"C0}{"DF}
```

Well, it would be nice if that were correct, but unfortunately, the Cork encoding contains some odd slots whose uccode or lccode isn't quite what you'd expect.

```
\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
\lccode`\^^9d=`\i    % dotted I
\uccode`\^^9d=`\^^9d % dotted I
\lccode`\^^9e=`\^^9e % d-bar
\uccode`\^^9e=`\^^d0 % d-bar
```

Finally here is one that helps hyphenation in the OT1 encoding.

```
\lccode`\^^[=`\^^[   % oe in OT1
\fi % End of reset block for 8-bit engines
```

`\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}%
```

Tell UTF-8 processing to process chars even though we are in an `\protected@edef`.

```
      \let\UTF@two@octets@noexpand\@empty
      \let\UTF@three@octets@noexpand\@empty
      \let\UTF@four@octets@noexpand\@empty
      \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}%
      \let\UTF@two@octets@noexpand\@empty
      \let\UTF@three@octets@noexpand\@empty
      \let\UTF@four@octets@noexpand\@empty
      \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}}
```

(*End definition for* `\MakeUppercase`, `\MakeUppercase`, *and* `\@uclclist`. *These functions are documented on page* **??**.)

1.8 Applying Patch files

Between major releases, small patches will be distributed in files `ltpatch.ltx` which must be added at this point.

Patch file code removed.

```
%\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%
%         !!!!
```

```
\toks6{}
\toks8{}
```

(*End definition for* `\toks`*. This function is documented on page* **??**.)

`\errhelp` Empty the error help message, which may have some rubbish:

```
\errhelp{}
```

(*End definition for* `\errhelp`*. This function is documented on page* **??**.)

1.10 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}
```

(*End definition for* `\@providesfile`*. This function is documented on page* **??**.)

`\@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}}%
```

(*End definition for* `\@filelist` *and* `\@addtofilelist`*. These functions are documented on page* **??**.)

1.11 Do some temporary work for pre-release

This is a good place to load code that hasn't yet been integrated into the other files ...

1.12 Some last minute initializations ...

Load the first aid set of definitions for external packages that await updates.

```
\@input{latex2e-first-aid-for-external-files.ltx}
```

1.13 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⟩
```

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. 490
1989-04-10 ltfssbas.dtx v1.0a
General: Starting with version numbers! `\ifmmode` added in `\math@group` 268
1989-04-10 ltfssbas.dtx v1.0b
General: `\preload@sizes` added. . . 268
`\wrong@fontshape` changed to define substitution font/shape macro. 268
1989-04-10 ltfssini.dtx v1.0a
General: Starting with version numbers `\newif` for `\@tempswa` added since this switch is unknown at the time when this file is read in. (latex.tex is loaded later.) `\math@famname` changed to `\math@version`. 367
1989-04-14 ltfssbas.dtx v1.0c
General: More documentation added. 268
1989-04-15 ltfssini.dtx v1.0b
General: `\mathfontset` renamed to `\mathversion`. 367
1989-04-19 ltfssbas.dtx v1.0d
General: Even more doc. 268
1989-04-21 ltfssbas.dtx v1.0e
General: Documentation is fun! Parameters of `\define@mathalphabet` changed. 268
1989-04-21 ltfssini.dtx v1.0c
General: Changed to conform to fam.tex. 367
1989-04-23 ltfssbas.dtx v1.0f
General: % in `\getanddefinefonts` added. 268
1989-04-26 ltfssini.dtx v1.0d
General: `\xpt` added. 367
1989-04-27 ltfssbas.dtx v1.0g
General: Documentation revised. . . 268
1989-04-27 ltfssini.dtx v1.0e
General: Definitions of LaTeX symbols corrected. 367
1989-04-29 ltfssbas.dtx v1.0h
General: Documented problem with `\halign`, and `\noalign` 268
`\mathversion`: Test if version defined added. 278
1989-04-29 ltfssbas.dtx v1.0i
General: Removed the `\halign` `\noalign` correction (wasn't bugfree) 268
1989-04-29 ltfssini.dtx v1.0f
General: Corrections to LaTeX tabular env. added. 367
1989-05-01 ltfssbas.dtx v1.0j
General: Default for `\baselinestretch` added. 268
1989-05-22 ltfssbas.dtx v1.0k
General: Lines longer than 72 characters folded. 268
1989-05-22 ltfssini.dtx v1.0g
General: Lines shortened to 72 characters 367
1989-09-14 ltfssbas.dtx v1.0m
General: Global replacement: `\group` to `\mathgroup` 268
`\mathversion`: Corrected typo: `\endscname` to `\endcsname`. 278
1989-11-07 ltfssini.dtx v1.0i
General: All family, series, and shape names abbreviated. 367
1989-11-08 ltfssbas.dtx v1.0o
General: First parameter of `\define@mathalphabet` and `\define@mathgroup` changed from string to control sequence. 268
1989-11-14 ltfssbas.dtx v1.0p
`\math@version`: Math version prefix 'mv@' added. 278
1989-11-19 ltfssbas.dtx v1.0q
`\define@newfont`: Group added. . . 280
`\wrong@fontshape`: Instead of calling `\family\default@family`, etc. we directly set `\f@family`, etc. 285
1989-11-22 ltfssbas.dtx v1.0r
`\math@version`: `\def` → `\edef` for `\math@version`. 278
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. 268
extra// → + in `\extra@def`. . . . 268

1989-11-26 ltfssbas.dtx v1.0t
\select@group: \bgroup/\egroup changed to \begingroup/\endgroup to avoid empty Ord atom on math list. . . 288
1989-12-02 ltfssini.dtx v1.1b
General: \rmmath renamed to \mathrm 367
1989-12-03 ltfssini.dtx v1.1c
General: Some internal macros renamed to make them inaccessible. 367
1989-12-05 ltfssbas.dtx v1.0u
\addto@hook: \addto@hook added. . 292
1989-12-05 ltfsstrc.dtx v1.0u fam.dtx
\every@math@size: Hook \every@size added. 319
1989-12-13 ltfsstrc.dtx v1.0f
\use@mathgroup: \expandafter added before final \fi. 322
1989-12-16 ltfssbas.dtx v1.1a
\select@group: \relax in front added. 288
Now four arguments. 288
Redefinition of alphabet now simpler. 288
Usage of '=' macro added. 288
1989-12-16 ltfsstrc.dtx v1.1a
\selectfont: Changed order of calls. 316
\use@mathgroup: Redefinition of alphabet now simpler. 322
Usage of '=' macro added. 322
1990-01-18 ltfsstrc.dtx v1.0h
General: \tracingfonts meaning changed. 312
1990-01-20 ltfssbas.dtx v1.2a
\math@bgroup: Def. placed in this file. 290
\math@egroup: Def. placed in this file. 290
\select@group: Def for alph id changed. 288
1990-01-21 ltfssbas.dtx v1.2b
\select@group: Code moved to \use@mathgroup. 288
1990-01-21 ltfsstrc.dtx v1.2b
\use@mathgroup: Macro added to allow cleaner interface. 322
1990-01-23 ltfssbas.dtx v1.2c
General: \no@version@warning renamed to \no@alphabet@error. 268
Macro \no@alphabet@help added 268
\no@alphabet@error: Changed to error call 268
1990-01-25 ltfssini.dtx v1.1e
\nfss@text: Macro added. 386
1990-01-27 ltfssbas.dtx v1.2d
\DeclarePreloadSizes: Font identifier set to \relax. 274
1990-01-28 ltfssbas.dtx v1.2e
\mathgroup: \newfam let to \new@mathgroup. 268
1990-01-28 ltfssbas.dtx v1.2f
\define@newfont: Added call to \curr@fontshape macro to allow substitution. 281
\wrong@fontshape: Warning message slightly changed. 285
1990-01-28 ltfssini.dtx v1.2b
\em: Call to \@nomath added. 384
1990-02-08 ltfssini.dtx v1.1g
General: Protected the commands \family, \series, \shape, \size, \selectfont, and \mathversion. 367
1990-02-16 ltfssbas.dtx v1.2g
General: Support for changes of \baselineskip without changing the size. 268
\math@version: \@nomath added. . . 278
1990-02-16 ltfsstrc.dtx v1.0i
\selectfont: Changed \f@size to \lcl@currsize (see fam file). . . 316
1990-02-18 ltfsstrc.dtx v1.0j
General: Redefine unprotected version \p@selectfont instead of \selectfont. 316
1990-03-14 ltfsstrc.dtx v1.0k
General: Added code for TeX3. 312
\extract@font: Added code for TeX3. 315
\selectfont: Added code for TeX3. 316
1990-03-30 ltfssbas.dtx v1.2h
\math@egroup: Changed to have one arg. 290
1990-03-30 ltfsstrc.dtx v1.2h
\use@mathgroup: Third argument removed (see \math@egroup). . . . 322
1990-04-01 ltfssbas.dtx v1.2i
General: Code added from tracefnt.dtx. 268
Support for TeX3. 268
1990-04-01 ltfsstrc.dtx v1.0l
General: Part of code moved to fam.dtx. 312
\tracingfonts: Check if \tracingfonts already defined. . 313

1990-04-01 ltfsstrc.dtx v1.0o
\tracingfonts: Check if \tracingfonts defined removed again. 313
1990-04-02 ltfssini.dtx v1.1i
General: \input of files now handled by docstrip. 367
1990-04-05 ltfsstrc.dtx v1.0m
\selectfont: Call \tracingon only if \tracingfonts greater than 3. . 316
1990-05-05 ltfsstrc.dtx v1.0n
\selectfont: \tracingon with new syntax. 316
1990-06-23 ltfssini.dtx v1.1k
\nfss@text: Changed to \mbox. . . . 387
1990-06-24 ltfssbas.dtx v1.2j
\DeclarePreloadSizes: Missing percent added. 274
1990-06-24 ltfsstrc.dtx v1.0o
\baselinestretch: Moved to tracefnt.dtx. 319
\getanddefine@fonts: \Adding tracing code. 323
\Macro moved from fam.dtx. . . . 323
Adding debug code. 323
\use@mathgroup: Tracing code added. 322
1990-06-30 ltfssbas.dtx v1.2l
\showhyphens: Macro added. 291
1990-06-30 ltfsstrc.dtx v1.0p
\use@mathgroup: Added \relax after math group number. 322
1990-07-07 ltfsstrc.dtx v1.0q
\getanddefine@fonts: Group number added to tracing. 323
\math@egroup: Tracing code added. 323
\use@mathgroup: Group number added to tracing. 322
1990-08-27 ltfsstrc.dtx 1.0r
\type@restoreinfo: Some extra tracing info. 318
1990-08-27 ltfsstrc.dtx v1.0r
\getanddefine@fonts: Correcting missing name after \tracingon. . 323
1991-03-28 ltfssini.dtx v1.1m
\copyright: Extra braces added. . . 387
1991-03-30 ltfssini.dtx v1.2g
\newfont: Definition added. 385
\symbol: Definition added. 386
1991-07-24 ltmiscen.dtx LaTeX2.09
\@verbatim: Added \penalty\interlinepenalty to definition of \par so that \samepage works 475
1991-08-14 ltmath.dtx LaTeX2.09
\cases: (RmS) inserted extra braces around entry for NFSS 485
1991-08-14 ltpictur.dtx LaTeX2.09
General: (RmS) inserted extra braces around entry for NFSS 557
1991-08-14 ltthm.dtx LaTeX2.09
\@endtheorem: Moved \itshape after \item to make it work with NFSS 587
1991-08-26 ltfssini.dtx v1.1n
\reset@font: Macro introduced . . . 387
1991-08-26 ltmiscen.dtx LaTeX2.09
\@verbatim: \@@par added 475
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) 555
1991-08-26 ltplain.dtx LaTeX209
\tracingall: Added \errorcontextlines=\maxdimen, suggested by J. Schrod 31
1991-09-29 ltboxes.dtx LaTeX2.09
\@mpfootnotetext: (RmS) added \reset@font 523
1991-09-29 ltfloat.dtx LaTeX2.09
\@footnotetext: (RmS) added \reset@font 620
1991-09-29 ltmath.dtx LaTeX2.09
\@eqnnum: RmS: \reset@font added. 490
1991-09-29 ltsect.dtx LaTeX2.09
\@dottedtocline: (RmS) added \reset@font for page number . . 599
1991-10-17 ltcntrl.dtx LaTeX209
\@tfor: (Rms) \xdef replaced by \def (See FMi's array.doc) 160
1991-10-25 ltbibl.dtx LaTeX2.09
\@citex: added \reset@font, suggested by Bernd Raichle. . . . 626
1991-11-01 ltfloat.dtx LaTeX2.09
\footnote: (RmS) Added \let\protect\noexpand in \footnote, \footnotemark, and \footnotetext, since \xdef is used 620
1991-11-04 ltlists.dtx LaTeX2.09
\makelabel: (RmS) added default definition for \makelabel, to produce an error message. 509
1991-11-04 ltplain.dtx RmS
General: Removed \itemitem since never needed/useful with LaTeX. . 29

1991-11-06 ltbibl.dtx LaTeX2.09
\\@citex: added code to remove a leading blank 626
1991-11-13 ltbibl.dtx LaTeX2.09
\\@bibitem: Changed counter enumi to enumiv, as it says in the comment above 626
1991-11-21 ltfssini.dtx v1.1o
\\reset@font: Added extra braces for robustness. 387
Changed to protected version of macro. 387
1991-11-22 ltfloat.dtx LaTeX2.09
\\footnote: (RmS) Added \\let\\protect\\noexpand in \\@xfootnote, \\@xfootnotemark, and \\@xfootnotetext 620
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. 509
1991-11-27 ltfssbas.dtx v1.3a
General: All \\family, \\shape etc. renamed to \\fontfamily etc. . . . 268
1991-11-27 ltfssini.dtx v1.2a
General: All \\family, \\shape etc. renamed to \\fontfamily etc. . . . 367
1992-01-06 ltfssini.dtx v1.2c
General: added slitex code 367
1992-01-10 ltbibl.dtx LaTeX2.09
\\@bibitem: Changed \\c@enumiv to \\value of \\@listctr 626
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.) 490
1992-01-10 ltthm.dtx LaTeX2.09
\\@othm: (RmS) Check for existence of theorem environment 586
1992-01-14 ltbibl.dtx LaTeX2.09
\\@biblabel: removed \\hfill 628
1992-01-14 ltsect.dtx 0.0
\\@starttoc: (RmS) added \\immediate to \\openout as all \\write commands are also executed \\immediate 597
1992-02-26 ltbibl.dtx LaTeX2.09
\\@lbibitem: Added \\hfill to restore left-alignment of bibliography labels in alpha style 626
1992-03-18 ltdefns.dtx LaTeX209
General: (RMS) changed input channel from 0 to \\@inputcheck to avoid conflicts with other channels allocated by \\newread 72
1992-03-18 ltfloat.dtx LaTeX2.09
\\@xympar: (RmS) added \\global\\@ignorefalse 614
\\end@float: (RmS) changed \\@esphack to \\@Esphack 609
1992-03-18 ltlists.dtx 0.0
\\trivlist: RmS: added \\@nmbrlistfalse 505
1992-03-18 ltmiscen.dtx LaTeX2.09
\\begin: Changed \\@ignoretrue to \\@ignorefalse (as documented) 467
1992-03-21 ltfssini.dtx v1.2d
General: Renamed \\text to \\nfss@text to make it internal. . 367
1992-05-12 ltfssbas.dtx v1.3c
\\extract@alph@from@version: Macro added. 289
\\select@group: Added call to \\extract@alph@from@version. . 288
1992-07-26 ltfssbas.dtx v1.9a
\\curr@fontshape: 280
\\DeclareFontShape: Introduced \\DeclareFontShape 269
\\define@newfont: 280
\\math@fonts: 287
\\select@group: 288
\\split@name: Added splitting into \\f@encoding. 280
\\wrong@fontshape: 285
1992-07-26 ltfsstrc.dtx v2.0b
\\s@fct@: 331
\\s@fct@sub: documentation fixes . . 332
\\selectfont: 316
\\try@simple@size: 326
\\try@size@range: 329
\\use@mathgroup: 322
1992-08-14 ltbibl.dtx LaTeX2.09
\\@citex: added missing argument braces around \\hbox, found by Ed Sznyter 626
1992-08-14 ltboxes.dtx LaTeX209
\\endminipage: (RmS) replaced \\vskip-\\lastskip by \\unskip (proposed by FMi) 523
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) 626

1992-08-19 ltsect.dtx 0.0
\\@xsect: (RmS) corrected bug: stretch and shrink in argument to \\hskip previously not negated 593
1992-08-19 ltthm.dtx LaTeX2.09
\\@othm: (RmS) Changed error message to complain about undefined counter 586
1992-08-20 ltfssini.dtx v1.4b
\\@setsize: Added \\@currsize. . . . 386
1992-08-24 ltdefns.dtx LaTeX209
\\@ifnextchar: (Rms) \\@ifnextchar didn't work if its first argument was an equal sign. 95
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. 467
\\verb: Changed \\verb and \\@sverb to work correctly in math mode . . . 479
1992-08-25 ltsect.dtx LaTeX2.09
\\@sect: (FMi) replaced explicit setting of \\@svsec by call to \\@seccntformat 592
1992-09-18 ltlists.dtx LaTeX2.09
\\item: (RmS) Added warning if \\item is used in math mode 507
1992-09-18 lttab.dtx LaTeX2.09
\\@array: Changed \\par to \\@empty to avoid starting new row e.g. after \\hline 540
1992-09-19 ltfsstrc.dtx v2.0c
\\try@simple@size: 326
1992-09-21 ltfssini.dtx v1.4d
\\not@math@alphabet: Macro defined. 385
1992-09-22 ltfssbas.dtx v1.91a
General: Introduced \\tf@size for math size. 268
1992-09-22 ltfsstrc.dtx v2.1a
\\getanddefine@fonts: Introduced \\tf@size for math size. 323
1992-11-13 ltfssini.dtx v?
\\hexnumber@: Made expandable. . . . 386
1992-11-23 ltcounts.dtx LaTeX209
\\stepcounter: Replaced {} in \\stepcounter by \\begingroup \\endgroup to avoid adding an empty ord in math mode 260
1992-11-26 ltboxes.dtx LaTeX2.09
\\@mpfootnotetext: (RmS) added protection for \\edef 523
1992-11-26 ltfloat.dtx LaTeX2.09
\\@footnotetext: (RmS) added protection for \\edef 620
\\footnote: (RmS) Changed all to ‘def‘protect‘noexpand‘protect‘noexpand 620
1992-12-03 ltfssini.dtx v?
\\hexnumber@: Make it accept counters. 386
1993-03-08 preload.dtx v2.0b
General: Added 12pt preloads 411
1993-03-18 ltfssbas.dtx v2.0c
General: Changed all \\@tempdima in \\@tempdimb to avoid killing \\numberline 268
1993-03-18 ltfsstrc.dtx v2.1b
General: Changed all \\@tempdima in \\@tempdimb to avoid killing \\numberline 312
Changed all \\@tempdimb in \\@tempdimx to avoid killing \\numberline 312
1993-03-18 ltfsstrc.dtx v2.1c
\\DeclareSizeFunction: Added all args to avoid blanks problems . . 329
1993-04-09 lterror.dtx v1.0e
\\@latexerr: Mention The Companion 166
1993-04-11 lterror.dtx v1.0f
\\@latexerr: Remove setting of errorcontextlines 166
1993-05-05 ltfntcmd.dtx v2.0b
General: Removed all LaTeX related cmds 414
1993-05-16 ltfssbas.dtx v2.0e
\\showhyphens: Use \\reset@font . . 291
1993-07-16 ltfsstrc.dtx v2.1h
General: Changed layout of info messages 312
1993-07-17 ltoutenc.dtx 1.0d
General: changed \\catcoding @ . . . 216
1993-08-03 ltmiscen.dtx LaTeX2.09
\\enddocument: Changed redefinition of \\global to redefinition of \\@setckpt. 461
1993-08-05 ltpictur.dtx LaTeX2.09
\\circle: (RMS) Added error message if \\circle is used in math mode. 577
1993-08-05 ltsect.dtx LaTeX2.09
\\@sect: (RmS) Made sure that \\protect works correctly in expansion of \\the counter 592

1993-08-05 ltspace.dtx LaTeX2e
\\@hspace: (RmS) Removed superfluous \\leavevmode in \\@hspace and \\@hspacer, as suggested by CAR. 190
1993-08-05 lttab.dtx latex2e
\\tabular*: Replaced \\expandafter\\def by \\@namedef. 539
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. 626
1993-08-13 ltoutenc.dtx 1.0f
General: Protected against active @ sign. 216
1993-08-13 preload.dtx v2.0c
General: Added \\relax at end of font names. 412
1993-08-16 ltoutenc.dtx 1.0g
General: Needs space after \\string 216
1993-08-18 ltfssdcl.dtx v2.0e
\\new@mathversion: Exchanged names of encodings in warning message of \\SetSymbolFont. 349
1993-09-02 ltfsstrc.dtx v2.1i
General: Corrected name of sgen size function. 312
1993-09-03 ltmiscen.dtx LaTeX2.09
\\verbatim@nolig@list: Replaced \\@noligs by extensible list 480
1993-09-07 ltmiscen.dtx LaTeX2.09
\\verb@balance@group: (RmS) Changed definition of \\verb so that it detects a missing second delimiter. 479
1993-09-08 ltmiscen.dtx LaTeX2.09
\\enddocument: Added warning in case of undefined references. 461
1993-09-15 ltfssbas.dtx v2.0g
\\DeclareFontEncoding: Corrected: \\default@T to \\default@M. 272
1993-09-15 ltfsstrc.dtx v2.1j
General: Corrected spelling of \\noxpand. 312
1993-09-19 lterror.dtx LaTeX2.09
\\@invalidchar: (RmS) Error message for invalid input characters. 169
1993-11-02 ltmath.dtx LaTeX2.09
General: RmS: Corrected description of \\@eqnsel, moved \\@eqnsel accordingly and removed extra \\tabskip assignment. 490
1993-11-03 ltmath.dtx LaTeX2e
General: RmS: Initialized \\everycr to empty 490
1993-11-03 ltpictur.dtx LaTeX2.09
General: (RmS) changed \\halign to \\ialign to initialize \\tabskip and \\everycr 557
1993-11-11 ltfssini.dtx v2.1a
\\normalfont: Macro added 387
1993-11-11 ltfsstrc.dtx v2.2a
General: Option concept added for LaTeX2e 312
1993-11-14 ltclass.dtx v0.2a
\\@currext: Name changed from \\@currextension 638
\\@reset@ptions: macro added 660
\\AtEndDocument: Included extension in the generated macro name for package and class hooks. 660
\\documentstyle: Added \\RequirePackage \\@unusedoptionlist stuff. 650
\\load@onefilewithoptions: Moved resetting of \\default@ds, \\ds@ and \\@declaredoptions here, from the end of \\ProcessOptions. 654
\\NeedsTeXFormat: made more robust for alternative syntax for other formats. 652
\\ProcessOptions*: Optimise 'empty option' code. 648
Stop adding the global option list inside class files. 648
1993-11-14 ltdefns.dtx v0.2a
\\g@addto@macro: Made global 99
1993-11-15 ltclass.dtx v0.2b
\\documentstyle: Modified to match \\ProcessOption* 650
\\ProcessOptions*: Star form added. 648
1993-11-17 ltclass.dtx v0.2c
\\@@fileswith@pti@ns: Macro added 660
\\@badrequireerror: Macro added . 661
\\@twoloadclasserror: Macro added 662
\\CurrentOption: Name changed from \\@curroption 638
\\DeclareOption*: Error checking added 647
\\load@onefilewithoptions: Added trap for two \\LoadClass commands. 657
\\NeedsTeXFormat: Name changed from \\NeedsFormat 652
\\ProcessOptions*: restoring \\@fileswith@pti@ns added. . . . 648

1993-11-18 ltclass.dtx v0.2d
\documentstyle: Modified \RequirePackage stuff. 650
\ExecuteOptions: Use \CurrentOption not \reserved@a 649
\NeedsTeXFormat: \fmtname \fmtversion not \@... 652
1993-11-21 ltfiles.dtx LaTeX2e
\@missingfileerror: Stop infinite looping on \@er@ext 209
1993-11-21 ltmiscen.dtx v0.9a
\@verbatim: use \verbatim@font instead of \tt 475
\verb: Use \verbatim@font instead of \tt. 479
\verbatim@font: Macro added 476
1993-11-22 ltclass.dtx v0.2f
\@fileswithoptions: Made the default [] not [\@unknownversion] 653
\@ifl@ter: Added //00 so parsing never produces a runaway argument. 642
General: \@unknownversion removed 633
\load@onefilewithoptions: Made the initial version [] not [\@unknownversion] 654
1993-11-22 ltdefns.dtx LaTeX2e
\@minus: Macro added 71
\@plus: Macro added 71
\CheckCommand: Macro added 78
\providecommand: Macro added 78
1993-11-22 lterror.dtx LaTeX2e
\c@errorcontextlines: Macro added 166
1993-11-22 ltfiles.dtx LaTeX2e
\listfiles: Removed checking for \@unknownversion 211
1993-11-22 ltlength.dtx LaTeX2e
\@settodim: Macro added 266
\@settopoint: Macro added 267
\settodepth: Macro added 266
\settoheight: Macro added 266
1993-11-22 ltlogos.dtx LaTeX2e
\LaTeXe: Macro added 192
1993-11-23 ltclass.dtx v0.2g
\@use@ption: Name changed from \@executeoption 649
General: Various macros now moved to latex.tex. 637
Warnings and errors now directly coded. 637
1993-11-23 ltdefns.dtx LaTeX2e
\@argdef: Macro added 74
\@ifundefined: Redefined to remove a trailing \fi 93
\@newcommand: Macro added 74
\@newenv: Macro interface changed . . 77
\@xargdef: Macro interface changed . 74
\@yargd@f: Avoid \@?@? token 75
Macro interface changed 75
\newcommand: Macro reimplemented and extended 73
\renewcommand: Macro reimplemented and extended 76
\renewenvironment: Macro reimplemented and extended 77
\two@digits: Macro added 70
1993-11-23 ltoutput.dtx v0.1a
\paperheight: Register added 734
\paperwidth: Register added 734
1993-11-23 ltoutput.dtx v0.1c
\@enlargepage: Command added . . 778
\@kludgeins: Insert added 778
\@makecol: Command changed 745
\@specialoutput: Command changed 739
\enlargethispage*: Commands added 778
1993-11-24 ltfntcmd.dtx v2.1a
\maybe@ic@: Use \t@st@ic 419
\t@st@ic: Macro added 420
1993-11-24 ltfssini.dtx v2.1a
General: Removed \xpt stuff 387
1993-11-24 ltlogos.dtx LaTeX2e
\LaTeX: Macro changed 192
1993-11-28 ltclass.dtx v0.2h
\@twoclasseserror: Macro added . 662
General: Assorted commands now in the kernel removed. 637
Directory syntax checing moved to dircheck.dtx 637
Primitive filenames now terminated by space not \relax. 637
\endfilecontents: Don't globally allocate a write stream (always use 15) 662
1993-11-28 ltfiles.dtx LaTeX2e
\@missingfileerror: Use filename parser from dircheck 209
1993-11-29 ltoutput.dtx v1.0b
\@makecol: \@makespecialcolbox added 745
\@makespecialcolbox: Command added 747
1993-11-29 ltplain.dtx LaTeX2e
General: All accents in decimals; suggested by Paul Taylor 30
1993-11-30 ltoutput.dtx v1.0c
\fl@tracemessage: Commands added 780

1993-12-01 fontdef.dtx v2.1a
General: Update for LaTeX2e 390
1993-12-01 ltoutput.dtx v1.0e
\@reinserts: Command added . . . 747
1993-12-03 ltboxes.dtx v0.1a
\@argrsbox: macro removed 525
\@begin@tempboxa: macro added . . 514
\@end@tempboxa: macro added 514
\@iirsbox: redefined to support \height 525
\@imakebox: macro modified 514
\@irsbox: redefined to support \height 525
\@isavebox: color support 517
extra group 517
\@isavepicbox: extra group 517
\@makebox: default changed from x to c 514
\@makepicbox: macro modified 515
\@savebox: default c not x 517
\bm@b: macros added 514
\endlrbox: macro added 517
\fbox: extra group 518
\lrbox: color support 517
macro added 517
\makebox: modified 513
\mbox: extra group 514
\minipage: Redefined to support extra optional arguments 522
\newsavebox: Pass the whole of arg 1 to \@ifdefinable 516
\parbox: Redefined to support extra optional arguments 520
\raisebox: redefined to support \height 525
\sbox: color support 516
extra group 516
\set@color: color support 516
macro added 516
1993-12-03 ltclass.dtx v0.2i
\@cls@pkg: Name changed to avoid clash with output routine. 661
General: \@onlypreamble: Many commands declared. 637
Removed obsolete \@documentclass 637
1993-12-03 lterror.dtx v1.0b
\@latexerr: Set \c@errorcontextlines to -1 . . . 166
1993-12-03 ltfssini.dtx v2.1a
General: update for LaTeX2e 367
1993-12-04 ltfilehook.dtx v0.9b
\unqu@tefilef@und: Macro added . 689
1993-12-04 ltfiles.dtx v0.9b
\@iinput: Macro reimplemented . . . 208
\@input: Macro reimplemented . . . 209
\IfFileExists@: Macro added 206
\input: Macro reimplemented 208
1993-12-05 ltfloat.dtx LaTeX2e
\@dblfloatplacement: Command changed 611
\@xfloat: Command changed 605
1993-12-05 ltoutput.dtx v1.0f
\@addtobot: Command changed . . . 759
\@addtocurcol: Command changed 761
\@addtodblcol: Command changed 772
\@addtonextcol: Command changed 768
\@addtotoporbot: Command changed 760
\@boxfpsbit: Command added . . . 783
\@flcheckspace: Command added . 785
\@flsetnum: Command added 784
\@flsettextmin: Command added . 785
\@flstop: Commands added 781
\@flupdates: Command added . . . 786
\@fpsadddefault: Command added 782
\@getfpsbit: Command added . . . 783
\@opcol: Command changed 745
Hook added 745
\@outputpage: Command changed . 749
\@resethfps: Command added . . . 784
\@setfloattypecounts: Command added 783
\@setfpsbit: Command added . . . 784
\@shipoutsetup: Command added . 749
\@startcolumn: Command changed 755
\@startdblcolumn: Command changed 755
\@testfp: Command added 783
\@textfloatsheight: Commands added 782
\@topnewpage: Commands changed 737
\@tryfcolumn: Command changed . 756
\@writesetup: \@startpagehook added 749
\output: Command changed 739
1993-12-06 ltclass.dtx v0.2k
\ExecuteOptions: Preserve \CurrentOption. 649
1993-12-06 ltoutput.dtx v1.0f
\@specialoutput: Unboxing of 255 added to rescue writes 739
1993-12-06 ltoutput.dtx v1.0g
\@topnewpage: \@floatplacement placement bug fixed 737
1993-12-07 ltclass.dtx v0.2l
\ProvidesFile: Macro added 646

1993-12-07 ltclass.dtx v0.2m
\load@onefilewithoptions: Reset \CurrentOption 654
1993-12-07 ltoutenc.dtx 1.1
General: Protected all special characters with \string. 216
1993-12-07 ltoutenc.dtx v1.1
General: Made all character numbers decimal. 213
Removed a lot of equal signs and the like. 213
1993-12-08 ltboxes.dtx v0.1b
\@begin@tempboxa: Extra braces for color support (braces removed from other macros) 514
\@irsbox: fix typo 525
\@parboxto: \endgraf added due to extra group in \@begin@tempboxa 520
\lrbox: move \@endpefalse out of the inner group 517
1993-12-08 ltfntcmd.dtx v2.1b
General: Macros \rm, \bf and \sf moved to classes.dtx 422
1993-12-08 ltlists.dtx LaTeX2e
\@item: use \sbox to support colour 508
1993-12-08 ltspace.dtx LaTeX2e
\@bsphack: Command reimplemented 180
Command reimplemented; late birthday present for Chris 180
\@vbsphack: Command added 183
1993-12-09 ltboxes.dtx v0.1c
\@irsbox: fix another typo 525
1993-12-09 ltclass.dtx v0.2n
\documentstyle: input 209 compatibility file. 650
1993-12-09 ltfiles.dtx v0.9e
\document: Hook added 195
1993-12-09 ltmiscen.dtx v0.9e
\enddocument: Hook added 461
1993-12-10 ltoutenc.dtx v1.2
General: Added source code for t1enc.sty. 213
1993-12-11 ltfntcmd.dtx v3.0a
General: Complete reworking of all text commands, using just one creator function 414
italic correction now put in front of penalty before glue 414
newcommands replaced by defs . . 414
newfontswitch command corrected and changed 414
\DeclareTextFontCommand: Macro changed 416
\emph: Macro changed 417
\fix@penalty: Macro added 420
\maybe@ic: Macro name changed . . 419
\maybe@ic@: Macro and name changed 419
\sw@slant: Macro changed 420
\textup: Macros changed 417
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. 490
1993-12-11 ltoutenc.dtx v1.2a
General: Corrected for t1enc, math. 213
1993-12-11 ltsect.dtx LaTeX2e
\author: Added default 588
\title: Added default 588
1993-12-11 ltxref.dtx LaTeX2e
\@setref: Macro added 457
\pageref: Macro reimplemented . . . 457
\ref: Macro reimplemented 457
1993-12-12 ltoutput.dtx v1.0h
\@cflb: boxmaxdepth setting moved 753
defs changed to lets 753
\@cflt: name changed 753
\@doclearpage: defs changed to lets 744, 745
\@makecol: defs changed to lets . . . 746
\@resethfps: Warnings added: minimal 784
\@startdblcolumn: defs changed to lets 755, 756
\@topnewpage: braces removed 737
\@tryfcolumn: defs changed to lets . 756
\fl@tracemessage: Commands changed 780
1993-12-13 ltclass.dtx v0.2o
General: Removed setting \errorcontextlines (now in latex.tex) 637
\documentstyle: compatibility file now latex209.sty. 650
\usepackage: Fixed error handling . 652
1993-12-13 ltdirchk.dtx v0.2a
General: on the 'docstrip' pass, do not check openin path 10
\IfFileExists: Removed interactive prompting for current directory syntax 10
\strip@prefix: modified, name changed from \stripmeaning. . . . 5
1993-12-13 ltlists.dtx latex2e
\trivlist: Initialised \@itemlabel 505

1993-12-13 ltmiscen.dtx v0.9h
\@noligs: Readded \@noligs 480
\@verbatim: Readded \@noligs . . . 475
Removed optional argument of \item 475
center: Removed optional argument of \item 473
flushleft: Removed optional argument of \item 474
flushright: Removed optional argument of \item 474
1993-12-13 ltoutenc.dtx v1.2b
General: Corrected file name in driver code. 213
1993-12-13 lttab.dtx latex2e
\tabbing: Removed optional argument of \item 534
1993-12-14 ltoutput.dtx v1.0i
General: Section added to declare all parameters 792
1993-12-15 ltboxes.dtx v0.1d
\@iminipage: Changed default from 'c' to 's' 522
\@iparbox: Changed default from 'c' to 's' 520
\minipage: Changed default from 'c' to 's' 522
extra space removed. 522
\parbox: Changed default from 'c' to 's' 520
1993-12-15 ltclass.dtx v0.2p
General: Removed extra '.'s from \@@warnings 637
1993-12-16 ltlogos.dtx LaTeX2e
\LaTeXe: Extended logo by DPC . . 192
1993-12-16 ltmath.dtx v0.9i
\@@eqncr: use \refstepcounter instead of shortcut 492
General: use \refstepcounter instead of shortcut 490
1993-12-16 ltmiscen.dtx v0.9i
General: \literal added 480
1993-12-16 ltpage.dtx LaTeX2e
\mark: Init \mark at begin document 631
1993-12-16 ltspace.dtx LaTeX2e
\@bsphack: Corrected optimisation :-) 180
1993-12-16 lttab.dtx latex2e
\@xhline: Measure from middle of vertical rules 549
1993-12-17 ltclass.dtx v0.2q
\@documentclasshook: Macro added 637
\@fileswithoptions: Add \@compatibility hook 653
\documentstyle: Match Alan's new code. 650
1993-12-17 ltoutenc.dtx 1.3
General: Added this section 217
Removed all the hackery for use in \DeclareFontEncoding, and redid everything using \DeclareTextFoo. 229, 231
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. . . . 216
1993-12-17 ltoutenc.dtx v1.3
General: Added \EncodingSpecificAccent, \EncodingSpecificAccentedLetter and \EncodingSpecificCommand. 213
Made Rokicki's encoding a proper encoding scheme rather than a variant of OT1. 213
1993-12-17 ltoutput.dtx v1.0j
\@opcol: Hook removed 745
\@specialoutput: Page room test added 740
\@topnewpage: check for vsize too small added 737
Page room test added 739
\@writesetup: —and then removed 749
\fl@tracemessage: tracefloatvals made a document command . . . 780
1993-12-17 ltpage.dtx LaTeX2e
\mark: Removed init \mark at begin document, since it doesn't work. 631
\rightmark: Stopgap solution to mark \leftmark and \rightmark work without initializing mark until the problem is solved. 631
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. 216
Replaced the missing last argument to \DeclareFontEncoding. 229, 231
1993-12-18 ltoutenc.dtx 1.3c
General: Rewrote for the new syntax of \EncodingSpecific. . . . 229, 231

Split \EncodingSpecificAccent up into \EncodingSpecific and \DeclareAccent. 217
1993-12-18 ltoutenc.dtx v1.3a
General: Replaced OT3 by XXX . . 213
1993-12-18 ltoutenc.dtx v1.3b
General: Corrected typos. 213
Replaced the missing last argument to \DeclareFontEncoding. 213
1993-12-18 ltoutenc.dtx v1.3c
General: A new syntax, separating accent-definitions from encoding-specific definitions, and allowing encoding-specific \chardef, \let, etc. 213
Rewrote for the new syntax of \EncodingSpecific. 213
1993-12-18 ltoutenc.dtx v1.3d
General: Some T1 stuff had drifted into the OT1 file. 213
1993-12-18 ltpage.dtx LaTeX2e
\sloppy: Added \emergencystretch 631
1993-12-19 ltclass.dtx v0.2r
\endfilecontents: Different message when ignoring a file 662
1993-12-19 ltfntcmd.dtx v3.0b
General: \@pdef command added . . 414
Added by ASAJ. 422
Made \@newfontswitch produce an error if command already exists, and added \@renewfontswitch, ASAJ 414
Other tidying 414
Some more tidying done 414
Untidying added, so this is now a TEMPORARY version. 414
Wording changes by CAR. 422
\DeclareOldFontCommand: Corrected and tidied 421
\DeclareTextFontCommand: Corrected and tidied 416
1993-12-19 ltspace.dtx LaTeX2e
\@bsphack: There seem to be problems with selfmade birthday presents . 181
1993-12-20 ltdefns.dtx LaTeX2e
\@reargdef: Kept old version of \@reargdef, for array.sty 75
1993-12-20 ltfiles.dtx v0.9m
\@obsoletefile: Added this command, removed @oldfilewarning 211
1994-01-05 fontdef.dtx v2.1d
General: Removed nf prefix from file names. 392
1994-01-13 ltmath.dtx v0.9o
\@@eqncr: correcting 0.9i 492
General: correcting 0.9i 490
1994-01-14 ltdirchk.dtx v0.2d
\IfFileExists: Close the texsys.aux output stream 10
1994-01-15 ltfiles.dtx v0.9o
\document: move \@preamblecmds after document hook 197
1994-01-17 ltclass.dtx v0.2s
\@fileswithoptions: Modify to reduce parameter stack usage . . 653
General: Added many more \@onlypreamble commands 637
Wrapped long lines to column 72 637
\load@onefile@withoptions: Modify to reduce parameter stack usage 657
1994-01-17 ltfiles.dtx LaTeX2e
\listfiles: New Version, adds '.tex' if needed, and lines up columns . 211
1994-01-17 ltfssbas.dtx v2.1a
General: New math font setup 268
\curr@math@size: New math font setup 279
\everydisplay: New math font setup 279
\everymath: New math font setup . 279
\frozen@everydisplay: New math font setup 279
\frozen@everymath: New math font setup 279
\math@version: New math font setup 278
1994-01-17 ltfssini.dtx v2.1e
\not@math@alphabet: Message changed 385
1994-01-17 ltfsstrc.dtx v2.3a
General: New math font setup 312
\check@mathfonts: New math font setup 321
\glb@currsize: New math font setup 318
\restglb@settings: New math font setup 321
1994-01-18 ltbibl.dtx LaTeX2e
\bibliography: Use \@input@ so include files are listed. 627
1994-01-18 ltclass.dtx v0.2t
\@ifclassloaded: Fix typo \@pkgetension 641
1994-01-18 ltfilehook.dtx v0.9p
\unqu@tefilef@und: New Definition 689
1994-01-18 ltfiles.dtx v0.9p
\@iffileonpath: Macro added 207
\@input: do not use a different definition for \input@path 209
\@input@: Macro added 209

\IfFileExists@: New Definition . . 206
\includeonly: Use \@input@ so include files are listed. 201
1994-01-18 ltfssini.dtx v2.1f
\not@math@alphabet: Message corrected 385
1994-01-18 ltmiscen.dtx v0.9p
\@verbatim: Add \global\@inlabelfalse 475
Only add \penalty if in hmode . 475
1994-01-19 fontdef.dtx v2.1e
General: Added missing setting for symbols in bold version. 397
1994-01-19 ltdirchk.dtx v0.2e
\IfFileExists: name changed from \test 9
\input@path: No longer check that an empty group is in the path 10
\strip@prefix: name changed from \strip@meaning, to match NFSS. 5
1994-01-19 ltmath.dtx v1.0n classes
\mathindent: Deferred setting of \mathindent 493
1994-01-20 ltdirchk.dtx v0.2f
General: \@copytexsys and the texsys.new file removed 8
Modify all of ltxcheck 13
\IfFileExists: \@copytexsys removed 10
1994-01-21 ltclass.dtx v0.2u
\documentstyle: compatibility file now latex209.def. 650
1994-01-21 ltdirchk.dtx v0.2g
General: Improve documentation, reorganise docstrip module 1
\filename@parse: Minor changes, and add Mac version (:) 11
\today: Name changed from \stamp, to save memory 9
1994-01-21 ltfloat.dtx LaTeX2e
\@xfloat: Added missing percent characters. 605
1994-01-21 ltmiscen.dtx v0.9s
\verbatim@font: Removed unnecessary category code hackery. 476
1994-01-24 ltdirchk.dtx v0.2h
\IfFileExists: Stop testing once texsys.aux has been found 9
1994-01-24 ltpage.dtx LaTeX2e
\pagestyle: (DPC) Complain if pagestyle is undefined. 629
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 279
1994-01-25 ltmath.dtx LaTeX2e
\bordermatrix: Removed \p@renwd. 485
1994-01-26 ltfsstrc.dtx v2.3c
\check@mathfonts: Correct trace info placement 321
\restglb@settings: Correct trace info placement 321
1994-01-27 ltfntcmd.dtx v3.1a
\nocorrlist: Only ., used as default for cm fonts 421
1994-01-29 ltclass.dtx v0.2v
\@@unprocessedoptions: Macro added. 661
\load@onefile@withoptions: All options raise error if no \ProcessOptions appears 658
1994-01-31 ltdefns.dtx v0.2w
\g@addto@macro: Use toks register to avoid ‘hash’ problems 99
1994-01-31 ltfiles.dtx v0.9t
\document: set \@normalsize or \normalsize if necessary 196
1994-01-31 ltfntcmd.dtx v3.1b
General: \@normalsize no longer defined 414
1994-02-01 ltpage.dtx LaTeX2e
\pagestyle: (DPC) Modify to get nicer error message 629
\thispagestyle: (DPC) Modify to get nicer error message 630
1994-02-02 ltclass.dtx v0.2x
\load@onefile@withoptions: Only run the hook and options check if the file was loaded. 658
1994-02-03 ltoutput.dtx v1.0k
\@makespecialcolbox: correct mistakes in the documentation . . 748
1994-02-07 ltclass.dtx v0.2y
\@fileswithoptions: Run \@compatibility on the first class to start (not the first to finish) . 653
\@ifclasswith: Add extra ,s so ‘two’ is not matched with ‘twocolumn’ 643
\ProcessOptions*: Add extra ,s so ‘two’ is not matched with ‘twocolumn’ 648

1994-02-07 ltfssbas.dtx v2.1c
\DeclareFontEncoding: revert catcode settings earlier 272
\DeclareFontShape@: revert catcode settings earlier 269
1994-02-08 ltoutput.dtx v1.0k
\@makespecialcolbox: boxmaxdepth setting added 748
boxmaxdepth setting removed . . 747
General: Documentation and tasks tidied. 722
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. 637
\@fileswithoptions: Renamed \@compatibility to \@documentclasshook. ASAJ. . . 653
1994-02-10 ltfssbas.dtx v2.1d
\addto@hook: Made \addto@hook long. 292
1994-02-10 ltfsscmp.dtx v2.1d
\scan@@fontshape: scan away stuff after pt 336
1994-02-22 ltfssini.dtx v2.1g
General: Correct error message 388
1994-02-24 ltfssbas.dtx v2.1e
\DeclareFontShape: Separate restoration of catcodes for fd cmds 269
\define@newfont: Separate restoration of catcodes for fd cmds 281
\nfss@catcodes: Separate restoration of catcodes for fd cmds 281
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 513
Remove need for drv file 513
1994-03-02 ltclass.dtx v0.3a
General: Remove need for driver file 637
1994-03-03 ltboxes.dtx v0.1f
\@irsbox: Replaced a missing \else 525
1994-03-04 ltfloat.dtx v1.0a
General: Initial version, split from latex.dtx 601
1994-03-04 ltsect.dtx v1.0a
General: Initial version, split from latex.dtx 588
1994-03-04 lttab.dtx v1.0a
General: Initial version, split from latex.dtx 527
1994-03-04 ltvers.dtx v1.0a
General: Initial version, split from latex.dtx 34
1994-03-07 ltboxes.dtx v0.1a
\@mpfootnotetext: Extra group for colour 523
1994-03-07 ltboxes.dtx v1.0a
General: Unify format with other Kernel files 513
1994-03-07 ltdefns.dtx v1.0a
\@@italiccorr: Macro added 71
1994-03-07 ltfiles.dtx v1.0a
General: Initial version, split from latex.dtx 193
Long lines wrapped to 72 columns 193
1994-03-07 ltfinal.dtx v0.1a
General: Add code from the old dump.dtx 809
Initial version, split from latex.dtx 797
move code here from lhyphen.dtx 802
Remove oldcomments environment 797
use \InputIfFileExists not \IfFileExists 802
1994-03-07 ltfloat.dtx v1.0a
\@endfloatbox: (DPC) Extra group for colour 610
\@footnotetext: (DPC) Extra group for colour 620
\@xfloat: (DPC) Extra group for colour 606
1994-03-07 lthyphen.dtx v0.1c
General: move the 2ekernel code to ltfinal.dtx 795
1994-03-07 ltlength.dtx v1.0a
\@settodim: (DPC) Extra group for colour 266
1994-03-07 ltlists.dtx v1.0a
General: Initial version, split from latex.dtx 497
Long lines wrapped to 72 columns 497
1994-03-07 ltpage.dtx v1.0a
General: Initial version, split from ltherest.dtx 629
1994-03-07 ltpictur.dtx v0.1a
General: Initial version, split from latex.dtx 552
Long lines wrapped to 72 columns 552

1994-03-07 ltsect.dtx v1.0a
\@hangfrom: (DPC)Extra groups for colour 595
1994-03-07 lttab.dtx v1.0a
General: Long lines wrapped to 72 columns 527
1994-03-08 ltclass.dtx v0.3b
General: Modify driver code into 'new style' 637
1994-03-08 ltdirchk.dtx v1.0a
General: Reorganise driver module into 'new style' 1
1994-03-08 ltplain.dtx v1.0a
General: Remove need for a driver file. 14
1994-03-10 ltfssbas.dtx v2.2f
\math@egroup: Changed \begingroup/\endgroup to \bgroup/\egroup. 290
1994-03-11 ltfssdcl.dtx v2.1b
\DeclareSymbolFontAlphabet@: Added check against use of alphabet switch outside of math mode. 365
\SetMathAlphabet@: Changed parameter template in temporary macro to catch check add below. 354
1994-03-12 ltclass.dtx v0.3c
General: Change name from docclass to ltclass 637
\ProvidesFile: Add \wlog 646
\ProvidesPackage: Add \wlog 644
use \@gtempa 644
1994-03-12 ltdefns.dtx v1.0b
\@reargdef: New defn, in terms of \@yargdef 75
\@yargd@f: Name changed from \XXX@argdef 75
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 614
\@xympar: (DPC) Extra bgroup for colour 614
1994-03-12 ltplain.dtx v1.0b
General: Name changed from lplain. The end of an era 14
1994-03-12 ltplain.dtx v1.0e
General: Replaced remaining width, height, depth by LaTeX macro names to save tokens. 14
1994-03-13 ltcntrl.dtx v1.0c
\@tfor: (DPC) Add \@tf@r so a single group is correctly treated. 160
1994-03-13 ltfilehook.dtx v0.3b
\unqu@tefilef@und: Use new cmd \@addtofilelist 689
1994-03-13 ltfiles.dtx LaTeX2e
\@addtofilelist: Macro added . . . 211
\listfiles: Reset \@addtofilelist at begin document 212
1994-03-13 ltfssbas.dtx v2.1g
General: add 2ekernel module to omit repeated code 268
1994-03-13 ltfssdcl.dtx v2.1c
General: add 2ekernel module to omit repeated code 340
1994-03-14 ltboxes.dtx v1.0b
\@isavebox: Use \color@setgroup . 517
\@isavepicbox: Use \color@setgroup 517
\color@begingroup: macro added for colour support 516
\color@endgroup: macro added for colour support 516
\lrbox: Use \color@setgroup 517
\sbox: Use \color@setgroup 516
1994-03-14 ltfloat.dtx 1.0c
\@xympar: (DPC) Use \color@begingroup 614
1994-03-14 ltfloat.dtx v1.0c
\@endfloatbox: (DPC) Use \color@endgroup 610
\@footnotetext: (DPC) Use \color@begingroup, add \endgraf 620
\@savemarbox: (DPC) Use \color@begingroup 614
\@xfloat: (DPC) Use \color@begingroup 606
1994-03-15 ltfiles.dtx LaTeX2e
\@missingfileerror: Quit on x or X just like a real error 209
1994-03-15 ltfntcmd.dtx v3.2a
General: Adapted to mass formatting 414
Changed \/ to \@@italiccorr . . 414
Removed \@renewfontswitch . . . 414
Removed defs of short-forms and all sizes except \normalize 414
1994-03-15 ltoutput.dtx v1.0l
\@addtocurcol: Changed \addvspace to \vskip 763, 767
\@combinedblfloats: Removed boxmaxdepth setting. 754

\@makecol: \maxdepth changed to \@maxdepth 746
Removed boxmaxdepth setting. . 746
\@makespecialcolbox: Removed boxmaxdepth setting. 748
\@topnewpage: Corrected and amended warning message 738
Warning added: it should be improved 739
General: Added some warnings when page gets full of top floats. 722
Driver added and further tidying. 722
Removed duplicated code and corrected docstrip options. 722
Some boxmaxdepth settings removed. 722
1994-03-16 ltclass.dtx v0.3f
General: Add pkgindoc package . . . 678
1994-03-16 ltfiles.dtx LaTeX2e
\listfiles: Move this code directly into \document 212
1994-03-16 ltfiles.dtx v1.0c
\document: (DPC) directly add file list settings 197
1994-03-16 ltmiscen.dtx v1.0b
\@verbatim: Remove \global\@inlabelfalse again. . 475
1994-03-28 ltalloc.dtx v1.0d
General: Redefinition of 'new' allocations removed. 155
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) 169
General: Remove test for \inputlineno undefined. 166
1994-03-28 ltfiles.dtx v1.0d
\document: (DPC) Use \normalsize not \@normalsize 196
(DPC) remove \@normalsize check 196
1994-03-28 ltfloat.dtx v1.0b
\@caption: Use \normalsize not \@normalsize 604
General: Split further from ltherest.dtx 601
1994-03-28 ltlists.dtx v1.0b
General: Improve documentation . . 496
1994-03-28 ltmiscen.dtx v1.0c
General: Improve Documentation . . 460
1994-03-28 ltplain.dtx v1.0c
\newlanguage: Remove some \outer declarations. 17
1994-03-28 ltsect.dtx v1.0b
General: Split further from ltherest.dtx 588
1994-03-28 lttab.dtx v1.0b
General: Improve documentation . . 527
1994-03-28 ltthm.dtx v1.0a
General: Initial version, split from latex.dtx 584
1994-03-29 ltcounts.dtx v1.0c
General: Create file from parts of ltmiscen and ltherest. 258
1994-03-29 ltlength.dtx v1.0c
General: Create file ltcntlen from parts of ltmiscen and ltherest. 266
1994-03-29 ltmiscen.dtx v1.0d
General: Remove counter macros to ltcntlen 460
1994-03-29 ltpageno.dtx v1.0c
General: Create file ltcntlen from parts of ltmiscen and ltherest. 454
1994-03-29 ltxref.dtx v1.0c
General: Create file ltcntlen from parts of ltmiscen and ltherest. 455
1994-03-31 ltbibl.dtx v1.0a
General: Initial version of ltidxbib.dtx, split from ltherest.dtx 625
1994-03-31 ltidxglo.dtx v1.0a
General: Initial version of ltidxbib.dtx, split from ltherest.dtx 622
1994-04-09 ltcounts.dtx v1.0d
\@newctr: \@nocnterr now has counter name argument 259
\addtocounter: \@nocnterr now has counter name argument 259
\setcounter: \@nocnterr now has counter name argument 259
\stepcounter: Use \addtocounter to have name checked 260
1994-04-09 ltthm.dtx v1.0b
\@othm: Use standard counter error message (FMi) 586
1994-04-11 ltclass.dtx v0.3g
\endfilecontents: Add star form, dont write \endinput at the end of the file. 662
\ProvidesFile: Protect against weird catcodes. 646
1994-04-11 ltfssbas.dtx v2.1h
General: Added \defaultscriptratio and \defaultscriptscriptratio. ASAJ. 268
\defaultscriptratio: Macro added 290
\defaultscriptscriptratio: Macro added 290

1994-04-12 ltboxes.dtx v1.0c
General: Remove \@acci, now defined in ltplain.dtx 521
Remove \@dischyph, now defined in ltinit.dtx 521
1994-04-12 ltdefns.dtx v1.0g
\@dischyph: Define \@dischyph, was previously in ltboxes.dtx 97
1994-04-12 ltplain.dtx v1.0d
General: Define \@acci 30
1994-04-12 ltvers.dtx v1.0b
General: Have version info generated automatically. 34
1994-04-14 ltfntcmd.dtx v3.2b
General: Macros renamed to non-private forms, JB 414
\DeclareOldFontCommand: Renamed from \@newfontswitch 421
1994-04-15 ltboxes.dtx v1.0d
\@isavebox: Added missing procent character. 517
1994-04-17 ltcounts.dtx v1.0e
\@newctr: Use \@nocounterr instead of \@nocnterr 259
\addtocounter: Use \@nocounterr instead of \@nocnterr 259
\setcounter: Use \@nocounterr instead of \@nocnterr 259
1994-04-17 lterror.dtx v1.0h
\@nocounterr: New name for error message, old error message (without arg) kept 167
1994-04-17 ltthm.dtx v1.0c
\@othm: Use new std counter error message (FMi) 586
1994-04-18 ltfinal.dtx v0.1b
General: Initialise \textheight, \textwidth and page style 799
1994-04-18 ltfloat.dtx v1.0d
\@footnotetext: (DPC) Remove Colour support 620
\@savemarbox: (DPC) Remove Colour support 614
1994-04-18 ltfssbas.dtx v2.1i
General: Macro \no@alphabet@help removed again 268
\calculate@math@sizes: Changed message to log only 290
\no@alphabet@error: Use std LaTeX error macro 268
1994-04-18 ltfssdcl.dtx ???
\DeclareMathAlphabet: Pass correct arg (2 not 3) 352
1994-04-18 ltfssdcl.dtx v2.1d
General: Removed surplus \no@alphabet@error (see fam.dtx) 340
1994-04-18 ltfsstrc.dtx v2.3d
General: Changed to new error/warning scheme 312
\font@submax: Changed dimen to macro 330
\fontsubfuzz: Changed dimen to macro 330
\subst@size: \font@submax and \fontsubfuzz now macros 331
1994-04-19 ltpage.dtx v1.0b
General: Improve documentation . . 629
1994-04-20 ltfntcmd.dtx v3.3a
General: Documentation up-dated . 414
New implementation of \nocorr . 414
\check@nocorr@: Macros added . . . 418
\maybe@ic@: \nocorr etc removed from list of tokens to check, leaving only punctuation characters 419
1994-04-20 ltmiscen.dtx v1.0e
\@enddocument@kernel@warnings: Changed logic for producing warning messages 463
1994-04-21 ltboxes.dtx v1.0e
\@iiiminipage: Extra \bgroup for colour 523
\@mpfootnotetext: Extra \endgraf for colour 523
\endminipage: Extra \egroup for colour 523
1994-04-21 ltfinal.dtx v0.1c
General: Added comments, set the catcodes of 128–255. 797
1994-04-22 ltfssini.dtx v2.1g
\not@math@alphabet: Message changed again 385
1994-04-23 ltfinal.dtx v0.1d
General: Check that \font@submax is still zero 797
1994-04-24 ltoutput.dtx v1.0m
\@resethfps: Number 2 changed to \tw@ 784
Warning changed 784
\@specialoutput: Message changed to give more info and 'top' removed 740
\@topnewpage: Message changed to give more info 739
Warning message removed as it will be generated later 738
General: Changed \@normalsize to \normalsize. 722

Corrected unverbed commands in documentation. 722
Removed some long lines and other aesthetic changes. 722
Warning messages changed/corrected. 722
1994-04-24 ltpictur.dtx v0.1b
General: Removed surplus spaces after `\hbox to` in several cases 552
1994-04-25 ltclass.dtx v0.3h
General: Removed spurious extra ‘.’s at the end of error messages 637
1994-04-25 ltfloat.dtx v1.0e
`\@largefloatcheck`: Changed warning message to give more info 610
Command added 610
General: Changed warning messages 601
Removed obsolete tracing code . . 601
1994-04-27 ltfsstrc.dtx v2.3e
General: Corrected item that was forgotten in last change. 312
1994-04-28 lterror.dtx v1.0j
`\@inmatherr`: Macro added 169
1994-04-28 lterror.dtx v1.1c
`\@inmatherr`: Replaced `\noexpand` with `\protect`. 169
1994-04-28 ltfssdcl.dtx v2.1e
General: Removed all `\uppercase` in hex num parsing macros 340
1994-04-28 ltlists.dtx v1.0c
`\item`: Replaced `\@ltxnomath` by `\@inmatherr` 507
1994-04-28 ltpictur.dtx v0.1c
`\@multiput`: (DPC) Macro added . . 556
General: bezier curves added 579
`\multiput`: (DPC) Ignore spaces between)(. 555
`\picture`: (DPC) Ignore spaces before (. 554
1994-04-28 ltplain.dtx v1.0g
General: Turn off overfull box tracing in log 25
1994-04-29 ltclass.dtx v1.0a
General: Change version number to 1 (no other change) 637
1994-04-29 ltmiscen.dtx v1.0f
`\@verbatim`: `\leavevmode` added . . 475
Change to `\everypar` added 475
1994-04-29 ltoutenc.dtx 1.4a
General: Removed `\EncodingSpecific`. Renamed all the commands. Added `\DeclareTextGlyph` and `\UndeclareTextCommand`. 217
Removed Rokicki’s OT1 variant encoding. Moved the driver to the top. 216
1994-04-30 ltfntcmd.dtx v3.3b
General: Documentation up-dated and tidied 414
Prefix frag@ changed to frag in `\@protecteddef` 414
Title changed 414
Warning changed to info message in `\@protecteddef` 414
1994-04-30 ltoutput.dtx v1.0n
`\@activechar@info`: `\@activechar@warning` changed to `\@activechar@info` 749
`\@combinedblfloats`: Removed rule in topnewpage case 754
`\@emptycol`: Empty column action added: `\@emptycol` 737
`\@flsetnum`: Rogue space removed . 785
`\@specialoutput`: Cut-off point changed to 2`\baselineskip` 740
Empty column action added: `\@emptycol` 740
Extra empty column added for twocolumn case 740
Extra empty column added for twocolumn case (wrong, see below) 740
`\@topnewpage`: Added setting of `\col@number` 737, 738
Cut-off point changed to 3`\baselineskip` 739
Empty column action added: `\@emptycol` 739
Message changed for Frank 739
General: `\@activechar@warning` changed to an info message. 722
Added `\col@number`. 722
Documentation tidied. 722
Empty column action added. . . . 722
Fixed bug from `\dblfigrule` with `\@topnewpage`. 722
Full of floats action improved. . . . 722
`\col@number`: Added `\col@number` . 734
`\onecolumn`: Added setting of `\col@number` 736
1994-05-01 lterror.dtx v1.0k
`\@latexerr`: (CAR) Added draft `\@latexinfo`. 166
1994-05-01 ltoutenc.dtx 1.4a
General: Added the `\a` command. . . 225
Added the `\SaveAtCatcode` and `\RestoreAtCatcode` commands. . 229

Removed the uc/lc table settings, since the T1 uc/lc table is now the default. 237
Rewrote for the new syntax. . 229, 231
1994-05-01 ltoutenc.dtx v1.4a
General: Removed Rokicki's encoding. 213
Renamed the commands, removed the `\EncodingSpecific` command. Turned all slots into decimal. Added `\a`. 213
1994-05-02 ltcntrl.dtx v1.0l
`\@break@tfor`: Macro added (from ltfiles.dtx) 160
1994-05-02 ltdefns.dtx v1.1f
`\renewcommand`: Removed surplus `\space` in error 76
`\renewenvironment`: Removed surplus `\space` in error 77
1994-05-02 ltfiles.dtx v1.0f
`\@iffileonpath`: `\@break@loop` renamed to `\@break@tfor` 207
`\@obsoletefile`: Make `\@onlypreamble` 211
1994-05-02 ltfinal.dtx v0.1e
General: Added setting the 'letter' catcodes. 807
Added setting the 'other' catcodes. 807
Added setting the special catcodes. 807
Made slot 127 illegal 807
Set all the catcodes 797
1994-05-02 ltfinal.dtx v0.1f
General: Set the catcode of control-J. 807
1994-05-02 ltmiscen.dtx v1.0g
General: Changed 91 to 1991 and moved some bits 460
1994-05-02 ltoutput.dtx v1.0o
`\@resethfps`: Code shortened 784
General: Code of `\@resethfps` shortened. 722
1994-05-03 ltbibl.dtx v1.0b
`\nocite`: Make `\nocite` issue a warning for an undefined citation key. 627
1994-05-03 ltfinal.dtx v0.1f
General: Set the catcode of control-J to be 'other', for use in messages. 797
1994-05-03 ltfloat.dtx v1.0f
General: (CAR) Added `\@largefloatcheck` 601
Removed unnecessary braces from arguments of `\@ifnextchar` 601
`\end@dblfloat`: `\@largefloatcheck` added 609
`\end@float`: (CAR) Added `\@largefloatcheck` 608
1994-05-03 ltfssdcl.dtx v2.1f
General: Renamed `\@@DeclareMathDelimiter` to `\@DeclareMathDelimiter` 340
1994-05-03 ltlists.dtx v1.0d
`\@item`: `\hskip` changed to `\kern` . . 508
`\item`: Removed superfluous braces 507
1994-05-03 ltmiscen.dtx v1.0h
`\@centercr`: `\@badcrerr` replaced by `\@nolnerr` 472
1994-05-03 lttab.dtx v1.0d
`\@endpbox`: Use `\@finalstrut` based on depth of `\@arstrutbox` 550
1994-05-04 ltclass.dtx v1.0b
`\NeedsTeXFormat`: Changed wording of the warning 652
1994-05-04 lterror.dtx v1.0m
`\@badcrerr`: Error message removed 169
1994-05-05 ltbibl.dtx v1.0c
`\@citex`: Set switch for warning and end of run. 626
`\nocite`: Do not write page number in `\nocite` warning message. 627
Set switch for warning and end of run. 627
1994-05-05 ltfinal.dtx v0.1g
General: Added empty errhelp. 797
`\errhelp`: Set error help empty. . . . 811
1994-05-05 ltfntcmd.dtx v3.3c
`\@@math@egroup`: Corrected `\@fontswitch` and added saved versions 421
General: Corrected `\@fontswitch` . . 414
1994-05-05 ltmiscen.dtx v1.0i
General: Removed braces from ifnextchar and ifstar arguments . 460
1994-05-07 lttab.dtx v1.0c
`\@maxtab`: Changed `\@firsttab` to `\chardef` 531
Changed `\@maxtab` to `\chardef` . 531
General: Removed definition of `\+` . 527
Removed surplus braces from `\@ifnextchar` constructs 527
1994-05-08 ltfntcmd.dtx v3.3d
General: Removed `\@undefinedfonterror` 414
`\normalsize`: Removed `\@undefinedfonterror` 422
1994-05-09 ltfntcmd.dtx v3.3f
General: Replaced all `\next` by `\@let@token` and undo change 3.3e, whatever that was. 414

1994-05-10 ltdefns.dtx v1.0n
General: (ASAJ) Added \DeclareProtectedCommand. 70
Added \DeclareProtectedCommand 79
Removed braces around \@ifundefined argument. ASAJ. 76
\makeatother: Added \makeatletter and \makeatother ASAJ. 97
1994-05-10 lterror.dtx v1.0n
\@latexerr: (ASAJ) Added extra blank lines to \@latexerr. 166
1994-05-10 ltmiscen.dtx v1.0j
\@@sverb: Slight change in error message text. 478
1994-05-11 ltboxes.dtx v1.0f
\@begin@tempboxa: Use new \color@setgroup concept. 514
\@iiiminipage: Use new \color@setgroup concept. 523
\@mpfootnotetext: Use new \color@setgroup concept. 523
Use new \normalcolor and \@finalstrut. 523
General: Superfluous braces removed from several commands 513
\color@setgroup: macro added for colour support 516
\endminipage: Use new \color@setgroup concept. 523
1994-05-11 ltclass.dtx v1.0c
\endfilecontents: Add checks for form feed and tab 662
1994-05-11 ltdirchk.dtx v1.0e
General: Add \ProvidesFile as used in fd files. 4
1994-05-11 lterror.dtx v1.0o
\@latexerr: (ASAJ) Removed one of the extra blank lines to \@latexerr. 166
1994-05-11 ltlogos.dtx v1.0o
\LaTeX: Use \DeclareProtectedCommand. ASAJ. 192
\LaTeXe: Use \DeclareProtectedCommand. ASAJ. 192
1994-05-11 ltoutenc.dtx 1.5a
General: Made T1 and OT1 generate packages rather than def files. Renamed the 'package' module to 'teststy'. 216
1994-05-11 ltoutenc.dtx v1.5a
General: Reimplemented \DeclareTextCommand using \@changed@cmd and \DeclareProtectedCommand. . . . 217
Renamed the commands again. Made the encoding part of the command syntax. Added the \DeclareTextCommand interface. Used \DeclareProtectedCommand. . . . 213
\DeclareTextAccent: Reimplemented using \DeclareTextCommand. . . . 219
1994-05-11 ltspace.dtx v1.0o
\hspace: Use \DeclareRobustCommand. ASAJ. 189
1994-05-12 ltboxes.dtx v1.0g
\@finalstrut: macro added 526
\fbox: New definition, merged with \framebox 518
\framebox: Merged \fbox and \framebox 518
\normalcolor: macro added for colour support 516
1994-05-12 ltdefns.dtx v1.0p
General: (ASAJ) Fixed a bug with \relax which was using \@gobble before defining it. 70
Fixed a bug with \relax which was using \@gobble before defining it. 79
1994-05-12 ltfssbas.dtx v2.1j
General: New baselinestretch concept 268
Replaced hand-protected commands by \DeclareRobustCommand defs 268
\f@linespread: New macro 278
\fontencoding: Use \DeclareRobustCommand. 276
\fontfamily: Use \DeclareRobustCommand. 277
\fontseries: Use \DeclareRobustCommand. 277
\fontshape: Use \DeclareRobustCommand. 277
\fontsize: Redefined to use \set@fontsize 277
\linespread: New macro 277
\mathversion: Use \DeclareRobustCommand. 278
1994-05-12 ltfssdcl.dtx v2.1g
General: Allow \relax as undefined command 340
Allow \relax'ed cmds to be declared 340
1994-05-12 ltfssini.dtx v2.1i
General: Moved \fontencoding to fam.dtx 367
Moved \fontfamily to fam.dtx . 367

Moved \fontseries to fam.dtx . 367
Moved \fontshape to fam.dtx . . 367
Moved \fontsize to fam.dtx . . . 367
Moved \mathversion to fam.dtx . 367
Moved \selectfont to tracefnt.dtx 367
1994-05-12 ltfsstrc.dtx v2.3f
\selectfont: Use \DeclareRobustCommand 316
1994-05-12 ltoutenc.dtx 1.5a
General: Removed the \SaveAtCatcode and \RestoreAtCatcode commands. . 229
Rewrote for the new syntax. . 229, 231
1994-05-12 ltoutput.dtx v1.0p
\@writesetup: \normalcoloradded 749
General: \normalcoloradded in various places (DPC). 722
1994-05-13 ltboxes.dtx v1.0h
\@arrayparboxrestore: New accent system, use \let not \def 521
1994-05-13 ltcounts.dtx v1.0f
General: Removed \@Ialph 262
Removed \@ialph 262
1994-05-13 ltdefns.dtx v1.0q
General: (ASAJ) Renamed \DeclareProtectedCommand to \DeclareRobustCommand. Removed \@if@short@command. . . 70
(ASAJ) Replaces \space by ‘ ’ in \csname. 70
Renamed \DeclareProtectedCommand to \DeclareRobustCommand. Removed \@if@short@command. Moved to after the definition of \@gobble. 79
1994-05-13 ltdefns.dtx v1.0r
General: (ASAJ) Added logging message to \DeclareProtectedCommand. 70
Added logging message to \DeclareProtectedCommand. 79
1994-05-13 ltdefns.dtx v1.0s
General: (ASAJ) Added \@backslashchar. 70
(ASAJ) Coded \@ifdefinable more efficiently. 70
Coded more efficiently, thanks to FMi. 76
1994-05-13 ltfiles.dtx LaTeX2e
\listfiles: Stop \listfiles being run twice 211
1994-05-13 ltfiles.dtx v1.0g
\document: Added execution of \every@size 196
1994-05-13 ltfinal.dtx v0.1h
General: Added package ot1enc, and defined \@acci, \@accii and \@acciii. 797
1994-05-13 ltfinal.dtx v1.0h
General: Added output enc stuff . . . 809
1994-05-13 ltfloat.dtx v1.0g
\@footnotetext: (DPC) Add new style colour support: \normalcolor 620
(DPC) Use \@finalstrut 620
\@xfloat: (DPC) Use \normalcolor 606
1994-05-13 ltfntcmd.dtx v3.3g
General: Replaced \@protecteddef by \DeclareRobustCommand 414
1994-05-13 ltfssbas.dtx v2.1k
General: Remove File identification ‘typeout’ 268
1994-05-13 ltfssbas.dtx v2.1l
\DeclareFontEncoding: Init encoding change command 272
\define@newfont: Use \@input@ for fd files 281
1994-05-13 ltfssdcl.dtx v2.1h
General: Removed file identification typeout 340
1994-05-13 ltfssini.dtx v2.1j
General: Removed file identification typeout 367
1994-05-13 ltfsstrc.dtx v2.3g
General: Removed typeouts as \ProvidesPackage writes to log. 312
1994-05-13 ltoutenc.dtx v1.5b
General: Added \{, \} and \$. 213
Renamed \DeclareProtectedCommand to \DeclareRobustCommand. 213
Replaces \space by ‘ ’ in \csname. 213
1994-05-13 ltpictur.dtx v0.1d
General: Removed surplus braces from \@if.. constructions 552
1994-05-13 lttab.dtx v1.0d
\@contfield: Colour support 533
\@startfield: Colour support 533
\@stopfield: Colour support 533
\a: moved to ltoutenc 531
1994-05-14 fontdef.dtx v2.1f
General: Removed .def files. 392
1994-05-14 ltfssbas.dtx v2.1m
\enc@update: Macro added 277

1994-05-14 ltfssbas.dtx v2.1n
General: Set defaults for all \f@... 277
\DeclareErrorFont: Don't set \f@encoding 284
\DeclareFontEncoding: Log if encoding is redeclared 272
Only init enc change cmd when new encoding 272
1994-05-14 ltfssini.dtx v2.1k
General: Init error font just before checking for fontdef.cfg 388
\reset@font: Remove surplus braces 387
1994-05-14 ltfsstrc.dtx v2.3h
\selectfont: Added \enc@update . 317
1994-05-14 ltoutenc.dtx 1.5d
General: Moved the driver to the top. 216
1994-05-14 ltoutenc.dtx v1.5c
General: Added the fontenc package 255
Added the fontenc package. 213
Fixed a bug which caused an infinite loop if \f@encoding was incorrectly set. 213, 217
Moved fontsmpl to its own dtx file. 213
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. . . . 213, 217
Tidied up the documentation. . . . 213
1994-05-14 ltoutenc.dtx v1.5e
General: Replaced \ENC@cmd by \ENC-cmd. 213
1994-05-15 ltfssbas.dtx v2.1o
General: encoding cmds changed to enc-cmd 268
1994-05-16 fontdef.dtx v2.1g
General: Removed \DeclareFontEncoding for ot1 and t1 and input .def files instead . . 392
1994-05-16 ltalloc.dtx v1.1a
General: (ASAJ) Split from ltinit.dtx. 155
1994-05-16 ltcntrl.dtx v1.0a
General: (ASAJ) Split from ltinit.dtx. 157
1994-05-16 ltdefns.dtx v1.1a
General: (ASAJ) Split from ltinit.dtx. 70
1994-05-16 lterror.dtx v1.1a
General: (ASAJ) Completely new error interface. 161
(ASAJ) Split from ltinit.dtx. . . . 161
1994-05-16 ltfinal.dtx v1.0i
General: moved output enc stuff to lfonts 809
1994-05-16 ltfssbas.dtx v2.1p
\fontsize: Pass \baselinstretch not \f@linespread 277
\linespread: Remove surplus braces 277
1994-05-16 ltfssini.dtx v2.1m
\@acciii: Define saved versions of accents 389
1994-05-16 ltlogos.dtx v1.1a
General: (ASAJ) Split from ltinit.dtx. 192
1994-05-16 ltmath.dtx v1.0k
\ensuremath: Use \DeclareRobustCommand and add extra braces in math mode 492
1994-05-16 ltoutenc.dtx 1.5h
General: \pounds was still using u rather than ui shape. 229
1994-05-16 ltoutenc.dtx v1.5f
General: enc files now have uc encoding name parts (FMi) 213
Revert code so that the encoding given is used in \DeclareTextCommand (FMi) . . . 213
1994-05-16 ltoutenc.dtx v1.5g
General: Made fontenc.sty use the new mixed-case encoding files. 213
Removed the lowercasing of the filename. 255
1994-05-16 ltoutenc.dtx v1.5h
General: Added \NG, \ng, \TH, \th, \DH, \dh, \DJ and \dj. 213
Added \r (ring accent) and \k (ogonek) accents. 213
Fixed a bug with \pounds. 213
Removed \P from the OT1 definitions file. 213
1994-05-16 ltoutenc.dtx v1.5i
General: Fixed a bug with \d. 213
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? 750
General: Changed setting of accents (FMi). 722
1994-05-16 ltpar.dtx v1.1a
General: (ASAJ) Split from ltinit.dtx. 171
1994-05-16 ltplain.dtx v1.0h
General: Comment out encoding specific commands 29
Remove \@acci and friends again . 30
Remove unnecessary def for \item 29
\loop: Use Kabelschacht method . . . 28
\m@th: Remove unnecessary space . . . 29

1994-05-16 ltspace.dtx v1.1a
General: (ASAJ) Split from ltinit.dtx. 173
1994-05-17 ltclass.dtx v1.0e
`\@use@ption`: Execute option after removing from list, not before . . 649
1994-05-17 ltdefns.dtx 1.1b
General: (ASAJ) Added the `\@protect@...` commands. 80
1994-05-17 ltdefns.dtx v1.1b
General: (ASAJ) Added definitions for protect. 70
(ASAJ) Removed warnings and logging to lterror.dtx. 70
Added the discussion of protected commands, defined the values that `\protect` should have. 80
1994-05-17 ltdefns.dtx v1.1c
General: (ASAJ) Redid definitions for protect. 70
1994-05-17 lterror.dtx v1.1b
General: (ASAJ) Moved error stuff from ltdefns.dtx. 161
1994-05-17 ltfssini.dtx v2.1n
`\copyright`: Really add extra braces 387
`\nfss@text`: Added braces to allow use in subscripts 387
1994-05-17 ltmath.dtx v1.0i
General: Replaced `\let` by `\gdef`, for indirect definition. 488
1994-05-17 ltoutenc.dtx v1.5j
General: Added braces to `\pounds` so it works as a subscript. 213
1994-05-18 ltdefns.dtx 1.1c
General: (ASAJ) Renamed the commands, and removed one which is no longer needed. 80
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. 80
1994-05-18 ltfinal.dtx v0.1j
General: Corrected the lccode for d-bar. 797
1994-05-18 ltlogos.dtx v1.1b
General: (ASAJ) Added the TEX logo. 192
(ASAJ) Made the LATEX 2ε logo use the text font '2' rather than the math font '2'. 192
1994-05-18 ltoutenc.dtx v1.5k
General: Made dotted-i produce 'i'. . 213
Removed braces from `\pounds` and `\dollar`. 213
Replaced `\defaultencoding` with `\encodingdefault`. 213
1994-05-19 ltbibl.dtx v1.1a
General: Initial version of ltbibl.dtx, split from ltidxbib.dtx 625
1994-05-19 ltcounts.dtx v1.1a
General: Extracted file from ltcntlen. 258
1994-05-19 ltdefns.dtx v1.1d
General: (RmS) Added definitions for `\@namedef` and `\@nameuse` again. . 70
1994-05-19 ltfinal.dtx v0.1k
General: Removed `\makeat...` 797
1994-05-19 ltidxglo.dtx v1.1a
General: Initial version of ltidxglo.dtx, split from ltidxbib.dtx 622
1994-05-19 ltlength.dtx v1.1a
General: Extract file ltlength from ltcntlen. 266
1994-05-19 ltpageno.dtx v1.1a
General: Extract file ltpageno from ltcntlen. 454
1994-05-19 ltplain.dtx v0.1k ltfinal
`\showoutput`: used `\maxdimen` not 99999 31
`\showoverfull`: used `\@ne` not 1 31
1994-05-19 ltxref.dtx v1.1a
General: Extract file ltxref from ltcntlen. 455
1994-05-20 ltdefns.dtx v1.1e
General: Changed command name from `\@checkcommand` to `\CheckCommand`. 70
`\CheckCommand`: Changed name from `\@checkcommand` to `\CheckCommand`. 78
1994-05-20 lterror.dtx v1.1c
General: (ASAJ) Added `\@latex@info@no@line`. 161
(ASAJ) Added missing full stops. 161
(ASAJ) Fixed a bug with `\@inmatherr`. 161
1994-05-20 ltfinal.dtx v0.1l
General: Use new font warning commands 803
1994-05-20 ltfloat.dtx v1.0h
`\@endfloatbox`: Restore outer value of @nobreak switch. 610
`\outer@nobreak`: Macro added: default is to do nothing. 610
1994-05-20 ltfntcmd.dtx v3.3h
General: Use new error commands . 414
1994-05-20 ltfssbas.dtx v2.1q
General: Use new error commands . 268

1994-05-20 ltfsstrc.dtx v2.3i
General: Use new error command names 312
1994-05-20 ltmiscen.dtx v1.0l
\@writefile: Added correct setting of \protect. 466
1994-05-20 ltmiscen.dtx v1.0m
General: Use new warning commands 460
1994-05-20 ltoutput.dtx v1.0s
\@writesetup: Added setting of \protect during \shipout. 749
General: Added setting of \protect during \shipout. 722
1994-05-20 ltpage.dtx v1.0d
\markright: Changed setting for \protect. 630
1994-05-20 ltsect.dtx v1.0c
General: Correct setting of \protect. 598
\addcontentsline: Correct setting of \protect. 597
1994-05-21 ltbibl.dtx v1.1b
General: Use new warning commands 625
1994-05-21 lterror.dtx v1.1d
General: (ASAJ) Made the error commands robust. 161
1994-05-21 ltfiles.dtx v1.0h
General: Use new error commands . 193
1994-05-21 ltlists.dtx v1.0f
General: Use new error commands . 496
1994-05-21 ltmiscen.dtx v1.0n
General: Use new error commands . 460
1994-05-21 ltsect.dtx v1.0d
General: Use new error commands . 588
1994-05-21 lttab.dtx v1.0f
General: Use new error commands . 527
1994-05-21 ltxref.dtx v1.1b
General: Use new warning commands 455
\newlabel: Use new warning commands 457
1994-05-22 ltclass.dtx v1.0f
General: Use new warning and error commands 633
1994-05-22 ltdefns.dtx v1.1f
General: Use new warning and error cmds 70
1994-05-22 lterror.dtx v1.1e
General: (ASAJ) Replaced bgroup by begingroup in error messages, to stop extra mathords creeping into math mode. 161
1994-05-22 lterror.dtx v1.2a
General: (ASAJ) Made \GenericError, \GenericWarning and \GenericInfo robust. 161
(ASAJ) Replaced \\ and tilde by \MessageBreak and \space. 161
(ASAJ) Replaced \@generic@message and \@generic@error by \GenericError, \GenericWarning and \GenericInfo. 161
(ASAJ) Replaces \string by \protect in some messages. . . . 161
1994-05-22 lterror.dtx v1.2d
\GenericError: (DPC) Alternative version added for old TeXs 162
(DPC) New version using long command name. 162
1994-05-22 ltfloat.dtx v1.0i
General: Use new warning commands 601
1994-05-22 ltoutput.dtx v1.0t
General: Changed warnings and infos to new commands. 722
1994-05-22 ltpictur.dtx v0.1e
General: Use new warning cmds . . . 552
1994-05-23 ltclass.dtx v1.0h
\NeedsTeXFormat: Don't stop completely when format is wrong 652
\usepackage: Remove argument if possible 652
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 329
1994-05-23 ltoutput.dtx v1.0u
\@activechar@info: Added \MessageBreak 749
\@writesetup: Changed resetting of \protect after shipout to use \aftergroup 749
General: Added \MessageBreak. . . . 722
Changed resetting of \protect after shipout. 722
1994-05-24 lterror.dtx v1.2e
\@latex@info@no@line: Macro added 165
1994-05-24 lterror.dtx v1.2f
General: (DPC) wrap long lines . . . 161
1994-05-24 ltfntcmd.dtx v3.3i
General: Tidying and typos fixed . . 414
1994-05-24 ltmiscen.dtx v1.0q
\@currenvline: Use \@empty as outer default 471
1994-05-25 ltdirchk.dtx v1.0g
\filename@parse: Mac parser had " typo for : 12

1994-05-25 ltfntcmd.dtx v3.3j
General: Insertion of \aftergroups to implement \nocorr moved to the end of the group 414
\check@icr: Macros added 418
\check@nocorr@: Insertion of \aftergroups moved and defaults set up for efficiency 418
\DeclareTextFontCommand: \expandafter inserted 416
Insertion of \aftergroups moved 416
1994-05-25 ltoutput.dtx v1.0v
General: Extra documentation. 722
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. 599
1994-05-25 ltthm.dtx v1.0c
General: Modify documentation . . . 584
1994-05-25 ltvers.dtx v1.0d
General: Remove PRELIMINARY TEST RELEASE from startup banner (spring is here) 34
1994-05-25 ltxref.dtx v1.1c
General: Modify documentation . . . 455
1994-05-26 ltfiles.dtx LaTeX2e
\@missingfileerror: Modify message format 209
1994-05-26 ltlogos.dtx v1.1c
General: Remove \SLiTeX logo 192
1994-05-26 ltmiscen.dtx v1.0r
General: \literal removed 480
1994-05-26 ltplain.dtx v1.1m
\iterate: (CAR) added \long 28
\underbar: (CAR/FMi) changed to use box \tw@ 29
1994-05-26 ltplain.dtx v1.1p
\underbar: (DPC) changed to use \sbox 29
1994-05-29 ltfssdcl.dtx v2.1j
General: Use new error commands . 340
1994-05-31 ltfinal.dtx v1.0n
General: Renamed lthyphen.* to lthyphen.*. 797
1994-06-01 ltboxes.dtx v1.0i
\@frameb@x: Macro added. 519
\@iframebox: New version, so \width is correct in \framebox 519
\fbox: New version, using \@frameb@x 518
\framebox: New version, so \width is correct in \framebox 518
1994-06-01 ltlogos.dtx v1.1d
\LaTeX: Add \m@th to force math size calculations 192
1994-06-01 ltoutput.dtx v1.0w
General: Tidied up typesetting. . . . 722
1994-06-08 ltfinal.dtx v1.0m
General: Add patch file system 809
1994-06-09 ltfinal.dtx v1.0n
General: For TEX2, do not set codes for higher half of character table. 801, 807
1994-06-09 ltfntcmd.dtx v3.3k
General: Tidying and typos fixed in documentation 414
1994-06-18 ltfntcmd.dtx v3.3l
General: Added check for empty text 414
\check@nocorr@: Added check for empty text 418
1994-06-22 ltfntcmd.dtx v3.3m
General: Removed space from \nfss@text 414
Renamed \check@nocorr 414
\check@nocorr@: Renamed \check@nocorr to \text@command to improve \long error message . 418
\DeclareTextFontCommand: Removed space from \nfss@text 416
1994-06-22 ltmath.dtx v1.2t classes
\mathindent: Set \mathindent at the end of the class instead of at begin document 493
1994-07-20 ltlogos.dtx v1.1e
\LaTeX: Save a few tokens 192
\LaTeXe: Save a few tokens 192
1994-07-20 ltpage.dtx v1.0h
\sloppy: Save a few tokens 631
1994-09-16 ltfssbas.dtx v2.1s
\nfss@catcodes: Reset [and] as well, just in case 282
1994-10-07 ltoutenc.dtx v1.5l
General: Moved the ogonek accent. . 213
1994-10-11 ltdirchk.dtx v1.0h
\@TeXversion: Check for TeX3.14 . . 13
General: Modify all of ltxcheck again 13
1994-10-12 ltsect.dtx v1.0f
General: Doc. typos 588
1994-10-14 fontdef.dtx v2.2a
General: New coding 390
1994-10-14 ltfssini.dtx v2.2a
General: New coding for cfg files . . . 367
1994-10-14 ltmiscen.dtx v1.0s
General: Move math to other file . . 460
1994-10-14 ltplain.dtx v1.1a
General: Moved code to other files. . . 14

1994-10-15 ltfssbas.dtx v2.1t
\extract@alph@from@version: Warn if math alpha is used outside math 290
1994-10-18 ltboxes.dtx v1.0j
\@frameb@x: \leavevmode added . . 519
\@iframebox: \leavevmode moved to \@frameb@x 519
\@parboxto: Macro added to remove misuse of \@empty 520
General: stuff from ltpatch done . . . 513
\fbox: \long added 518
\mbox: \long added 514
\sbox: \long added 516
1994-10-18 ltclass.dtx v1.0j
General: Move \listfiles to ltfiles.dtx 633
1994-10-18 ltdefns.dtx v1.2a
\@star@or@long: macro added 73
General: Add extra test for \endgraf 70
Add star-forms for all commands . 70
\renew@environment: reset end command 77
1994-10-18 ltfiles.dtx v1.0i
\listfiles: code moved here from ltclass 211
1994-10-18 ltoutenc.dtx v1.5l
General: Added new definitions of \patterns and \hyphenation. . . 225
1994-10-18 ltoutenc.dtx v1.5m
General: Added new definitions of \patterns and \hyphenation. . . 213
1994-10-18 ltsect.dtx v1.0g
\@dottedtocline: Added \normalcolor for page number . 599
General: Added \normalcolor 588
1994-10-19 ltfssbas.dtx v2.1t
\DeclareFontEncoding: Add missing \relax. 272
1994-10-23 ltfsstrc.dtx v23.k
\every@math@size: Renamed to \every@math@size 319
1994-10-23 ltmath.dtx v1.0l
\@eqnnum: Added \normalcolor since \eqno introduces a subgroup of the displayed math group 490
\ensuremath: Remove extra braces: but see p 168 of Leslie's book . . 492
1994-10-24 ltboxes.dtx v1.0k
\fbox: Inner braces added (to fix latex/1061) 518
1994-10-25 fontdef.dtx v2.2c
General: Added OMSenc.def 392
1994-10-25 ltboxes.dtx v1.0l
\@isavepicbox: missing percent (moved from ltpatch) 517
1994-10-25 ltdefns.dtx v1.2b
General: Documentation improvements 70
1994-10-25 ltoutenc.dtx 1.6a
General: Added \textdollar, \textlbrace, \textrbrace, \textsterling, \textunderline. 231
Removed \textlbrace, \textrbrace, \textunderline to give them their proper names. . . 231
1994-10-25 ltoutenc.dtx v1.6a
General: Added \ProvideTextCommand, \UseTextSymbol, \UseTextAccent, \DeclareTextSymbolDefault, \DeclareTextAccentDefault, \DeclareTextCommandDefault, and \ProvideTextCommandDefault. . 213
Added the \Provide commands, and the default definitions. 217
Added the defaults. 225
Added the files OT1enc.def, T1enc.def and OMSenc.def. 225
Added the OMS encoding. 237
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 231
Added: \textemdash \textendash \textexclamdown \texthyphenchar \texthyphen \textquestiondown \textquotedblleft \textquotedblright \textquoteleft \textquoteright 229
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. 217
1994-10-27 ltoutenc.dtx v1.6b
General: Added \textbackslash. . . 237
Added more defaults for OT1. . . 225
Removed the enc.def files 213
Removed the files OT1enc.def, T1enc.def and OMSenc.def. 225
Renamed \textlbrace to \textbraceleft and \textrbrace to \textbraceright. 237
1994-10-29 ltmath.dtx 1.0m
General: ASAJ: Added \DeclareMathOperator. 481
ASAJ: Tidied up documentation. 488
1994-10-29 ltmath.dtx v1.0m
General: ASAJ: Added \mathellipsis, \mathdollar and \mathsterling. 488
ASAJ: Removed \dag, \ddag. . . . 488
ASAJ: Renamed \S and \P to \mathsection and \mathparagraph and made them \mathchardefs. 488
1994-10-29 ltoutenc.dtx v1.6c
General: Added commands like \dots for use in text and math. 225
Renamed \P, \S, \dag and \ddag to \textparagraph, \textsection, \textdagger and \textdaggerdbl. 213
1994-10-30 ltdefns.dtx v1.2c
\@onelevel@sanitize: Macro added . 96
General: (CAR)\@onelevel@sanitize added 70
1994-10-30 ltdefns.dtx v1.2f
General: (DPC)\newwrite's moved to ltfiles 70
1994-10-30 ltmath.dtx v1.0n
General: ASAJ: Moved the new commands to ltoutenc. 488
1994-10-30 ltoutenc.dtx v1.6d
General: Added \DeclareTextCompositeCommand. 213
Added \textcircled. . 213, 227, 237
Added \t. 227
Added math commands. 213
Added OML encoding. 213, 227
Added the OML encoding. 237
Made \textless and \textgreater come from OML. 227
Moved math commands here from ltmath. 229
Removed \textregistered. 227
Rewrote \copyright to use \textcircled. 227
1994-10-31 fontdef.dtx v2.2d
General: Added OMLenc.def 392
1994-10-31 fontdef.dtx v2.2e
General: ... and moved further down 392
1994-10-31 ltfloat.dtx v1.1a
\@dblfloat: Major changes since two-column and one-column cases merged 604
\@dblflset: Macro added 604
Major changes to parameter parsing, setting of local variables, etc; two-column and one-column cases merged; space hacks moved 604
\@endfloatbox: (DPC/CAR) Extra box added to remove colour resetting from vmode 610
\@floatboxreset: Macro added . . . 608
\@footnotetext: (DPC/CAR) Move colour setting to output routine . 620
\@savemarbox: (DPC/CAR) Extra box added for colour 614
\@setfps: Macro added 605
\@xdblfloat: Macros removed: \@dbflt, \@xdblfloat 610
\@xfloat: (DPC/CAR) Extra box added to remove colour resetting from vmode 606
Major changes, removing setting of local variables, space hacks etc; two-column and one-column cases merged 605
Reset hook added 606
\@xympar: (DPC/CAR) Extra box added since needed for floats . . . 614
\fps@dbl: Macro added 605
1994-10-31 ltoutput.dtx v1.1a
\@makecol: (DPC/CAR) Colour resetting moved to here 746
\@topnewpage: (DPC/CAR) Extra box added to remove colour resetting from vmode 738
(DPC/CAR) Use \color@begingroup for colour . . 738
(DPC/CAR) Use \normalcolor . 738
1994-11-02 ltoutenc.dtx v1.6d
General: Wrapped lines longer than 70 characters. 213

1994-11-03 ltclass.dtx v1.0k
General: Move \@missingfileerror to ltfiles 637
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 209
1994-11-04 ltboxes.dtx v1.0m
\@mpfootnotetext: Added \protected@edef. ASAJ. 523
1994-11-04 ltdefns.dtx v1.2e
General: Added \set@display@protect to \typeout. ASAJ. 70
Added commands for setting and restoring \protect. ASAJ. 82
Rewrote protected short commands using \x@protect. ASAJ. 80
1994-11-04 lterror.dtx v1.2g
General: Added \set@display@protect to \Generic* commands. ASAJ. . . 161
1994-11-04 ltfiles.dtx v1.0k
\nofiles: Added setting of \protected@write, \makeindex and \makeglossary to \nofiles. ASAJ. 200
\protected@write: Macro added ASAJ. 201
1994-11-04 ltfloat.dtx v1.1b
\@footnotetext: (ASAJ) Added \protected@edef. 620
\footnotemark: Added \protected@xdef to \footnotemark. 620
1994-11-04 ltidxglo.dtx v1.1b
\@wrglossary: Added \protected@write to \@wrglossary. 624
\@wrindex: Added \protected@write to \@wrindex. 623
General: Removed \if@filesw from \makeindex. 622
\makeglossary: Removed \if@filesw from \makeglossary. 623
1994-11-04 ltmiscen.dtx v1.0t
\@writefile: Removed setting of \protect. ASAJ. 466
1994-11-04 ltoutenc.dtx v1.6f
General: Added _ 228
Added \mathunderscore. 229
1994-11-04 ltpage.dtx v1.0e
\markright: Added \@unexpandable@protect. ASAJ. 630
1994-11-04 ltsect.dtx 1.0h
\@sect: (ASAJ) Added \protected@edef. 592
General: (ASAJ) Added \protected@xdef to \thanks. . . 588
1994-11-04 ltsect.dtx v1.0h
General: Added \protected@write to \addtocontents. ASAJ. 598
\addcontentsline: Added \protected@write to \addcontentsline. ASAJ. 597
1994-11-04 lttab.dtx v1.0h
\@mkpream: (ASAJ) Added \@unexpandable@protect to \@mkpream. 546
\multicolumn: (ASAJ) added \set@typeset@protect. 542
1994-11-04 ltxref.dtx v1.1d
\label: (ASAJ)Added \protected@edef 457
(ASAJ)Added \protected@write 457
1994-11-05 ltboxes.dtx v1.0n
\@mpfootnotetext: Colour resetting for footnotes moved to endminipage: as for main page. . 523
\color@endbox: macro added for colour support 516
\color@hbox: macro added for colour support 516
\endminipage: Colour resetting for footnotes moved to here: as for main page. 523
1994-11-05 ltboxes.dtx v1.0o
\@mpfootnotetext: Colour groups restored here. 523
1994-11-05 ltfloat.dtx v1.1c
\@dblflset: Add compatibility with old version of \@xfloat. 604
\@endfloatbox: Use new \color@hbox concept. 610
\@footnotetext: Removed \normalcolor (again) 620
\@savemarbox: Use new \color@hbox concept. 614
\@setfps: Add compatibility with old version of \@xfloat. 605
\@xfloat: Add compatibility with old version of \@xfloat: but the arguments, provided at exorbitant cost, are now completely ignored 605
Use new \color@hbox concept. . . 606

\@xympar: Use new \color@hbox concept. 614
1994-11-05 ltoutenc.dtx v1.6g
General: Added setting of \@typeset@protect to \patterns and \hyphenation. 225
1994-11-05 ltoutput.dtx v1.1b
\@topnewpage: Use new \color@hbox concept. 738
\@writesetup: Change protect settings for new-style, protect-free aux-files. 749
Use new \color@hbox concept. . . 749
1994-11-05 ltoutput.dtx v1.1c
\@begindvi: Added macro 753
\@begindvibox: Added macro 734
\@writesetup: Add new \AtBeginDvi concept 749
\AtBeginDvi: Added macro 734
1994-11-06 ltfssbas.dtx v2.1u
\cf@encoding: New macro 278
\DeclareFixedFont: Renamed \every@size to \every@math@size. 270
1994-11-06 ltfssini.dtx v2.2b
\@setsize: Use \@typeset@protect 386
1994-11-06 ltfsstrc.dtx v2.3k
\glb@currsize: New implementation 318
\try@simples: New implementation 330
\try@size@substitution: New implementation 329
\tryis@simple: New implementation 330
1994-11-07 fontdef.dtx v2.2f
General: (DPC) Add \DeclareMathSizes declarations 397
(DPC) Updated to use \ProvidesFile 392
1994-11-07 ltfiles.dtx v1.0l
\@unused: move here from ltdefns, remove duplicate \@mainaux . . . 195
1994-11-07 ltfiles.dtx v1.0m
\document: Renamed \every@size to \every@math@size. 196
1994-11-07 preload.dtx v2.1e
General: (DPC) Updated to use \ProvidesFile 411
1994-11-09 ltboxes.dtx v1.0p
\@finalstrut: Revert \finalstrut to 2.09 equivalent (from ltpatch) . . 526
General: more colour changes... . . . 513
1994-11-09 ltfssbas.dtx v2.1v
\@vpt: (DPC) macros added, from setsize.dtx 292
(DPC) reduce save stack usage latex/1742 292
1994-11-10 ltbibl.dtx v1.1c
General: Fix \nocite{*} 625
\nocite: Fix \nocite{*} 627
1994-11-10 ltmath.dtx v1.2v classes
eqnarray: Added value of \parskip to \abovedisplayskip to compensate for negative \topsep 495
1994-11-10 ltoutput.dtx v1.1e
\@writesetup: Modify \protect setting 749
1994-11-10 ltplain.dtx v1.1b
General: (CAR) added patch to \loop. 14
\iterate: (CAR) added extra \relax 28
1994-11-11 ltspace.dtx v1.2a
\\: (DPC) Make robust 177
1994-11-12 ltfntcmd.dtx v3.3o
\normalsize: Added \MessageBreak 422
1994-11-12 ltlists.dtx v1.2b ltspace
\endtrivlist: Changed order of tests to make \@noitemerror correct: end of an era. 505
1994-11-12 ltmiscen.dtx v1.0u
center: Changed end macro to \def: safer and consistent 473
flushleft: Changed end macro to \def: safer and consistent 474
flushright: Changed end macro to \def: safer and consistent 474
1994-11-12 ltplain.dtx v1.1c
General: Comment out more encoding specific commands 29
1994-11-12 ltspace.dtx v1.2b
\addpenalty: Corrected error message 185
\addvspace: Corrected error message 184
1994-11-13 ltspace.dtx v1.2c
\addpenalty: Recorrected error message 185
\addvspace: Recorrected error message 184
1994-11-14 ltoutput.dtx v1.1f
\@begindvi: Use normal box register: why a box? 753
\@begindvibox: Use normal box register: why a box? 734
\@writesetup: Modify new \AtBeginDvi concept 749
General: Removed old definition of \@testfp. 722
1994-11-14 ltspace.dtx v1.2d
\\: (DPC) Macro modified 177

1994-11-14 lttab.dtx v1.0i
\tabularnewline: (DPC) Macro added 541
1994-11-16 fontdef.dtx v2.2h
General: (DPC) Removed \{ and \} 392
1994-11-17 ltboxes.dtx v1.0q
General: \@tempa to \reserved@a . . 513
1994-11-17 ltclass.dtx v1.0l
General: \@tempa to \reserved@a . . 633
1994-11-17 ltcntrl.dtx v1.0b
General: \@tempa to \reserved@a . . 157
1994-11-17 ltdefns.dtx v1.0g
General: \@tempa to \reserved@a . . . 70
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 . . 161
1994-11-17 ltfiles.dtx v1.0n
General: \@tempa to \reserved@a . . 193
1994-11-17 ltfinal.dtx v1.0o
General: \@tempa to \reserved@a . . 797
1994-11-17 ltfloat.dtx v1.1e
General: \@tempa to \reserved@a . . 601
1994-11-17 ltfntcmd.dtx v3.3p
General: \@tempa to \reserved@a . . 414
1994-11-17 ltfssbas.dtx v2.1w
General: \@tempa to \reserved@a . . 268
1994-11-17 ltfssdcl.dtx v2.1m
General: \@tempa to \reserved@a . . 340
1994-11-17 ltfsstrc.dtx v2.3l
General: \@tempa to \reserved@a . . 312
1994-11-17 ltmath.dtx v1.0o
General: \@tempa to \reserved@a . . 481
1994-11-17 ltmiscen.dtx v1.0v
General: \@tempa to \reserved@a . . 460
1994-11-17 ltoutenc.dtx v1.6h
General: (DPC) \@tempa to \reserved@a 213
1994-11-17 ltoutput.dtx v1.1h
General: \@tempa to \reserved@a. . 722
1994-11-17 ltpictur.dtx v1.0f
General: \@tempa to \reserved@a . . 552
1994-11-17 ltsect.dtx v1.0i
General: \@tempa to \reserved@a . . 588
1994-11-17 lttab.dtx v1.0j
General: \@tempa to \reserved@a . . 527
1994-11-18 ltboxes.dtx v1.0r
\color@vbox: macro added for colour support 516
1994-11-18 ltfinal.dtx v1.0n
General: re-allow slots 127–255 807
1994-11-18 ltfssbas.dtx v2.1x
General: (DPC) use \reserved@f not \next 268
1994-11-18 ltfssdcl.dtx v2.1m
\DeclareMathDelimiter: (DPC) \expandafter instead of \next . 359
1994-11-18 ltfsstrc.dtx v2.3m
General: \next to \reserved@f . . . 312
1994-11-18 ltmath.dtx v1.0p
\phantom: (DPC) colour support . . 483
(DPC) use \expandafter instead of \next 483
\prime@s: (DPC) use \@let@token instead of \next and \expandafter instead of \nxt . . 488
\smash: (DPC) colour support 484
(DPC) use \expandafter instead of \next 484
1994-11-21 ltfloat.dtx v1.1f
\@endfloatbox: Added reset of minipage flag 610
Corrected position of \outer@nobreak 610
\@marginparreset: Macro added . . 614
\@savemarbox: Added \@setminipage etc 614
Added resetting of size and font . 614
Changed to \color@vbox 614
Use \@setnobreak etc 614
\@setminipage: Macro added 608
\@setnobreak: Macro added 608
\@xfloat: Added \@setminipage . . 606
Added resetting of size and font . 606
Changed to \color@vbox so that large floats overflow at the bottom 606
Missing percents reinserted after 4, 8: these are not numbers. 605
Use \@setnobreak 606
\@xympar: Changed to \color@vbox 614
1994-11-21 ltoutput.dtx v1.1i
\@addtocurcol: Added \if@nobreak test before float box 763, 767
\@specialoutput: Added \if@nobreak test 742
\@topnewpage: Changed to \color@vbox 738
1994-11-22 ltfssdcl.dtx v2.1o
General: wrap long lines 340
1994-11-22 ltoutenc.dtx v1.6i
General: Corrected \dots so that there's no kerning in monowidth fonts. 213
Corrected typo with \mathunderscore. 213
Fixed empty accents. Again. . . . 213

1994-11-24 ltdefns.dtx v1.2h
\@newenv: Added test for \endgraf . . 77
1994-11-25 ltplain.dtx v1.1f
General: (DPC) Comment out lots of obsolete code 14
1994-11-26 ltfloat.dtx v1.1b
\footnote: (ASAJ) Added \protected@xdef. 620
1994-11-28 ltcntrl.dtx v1.0c
General: Documentation improvements 157
1994-11-30 ltfiles.dtx v1.0o
\@dofilelist: Macro added 212
\listfiles: Use \@dofilelist . . . 211
\nofiles: There is no \@gobblethree... 200
1994-11-30 ltfssbas.dtx v2.1y
\fontshape: Use \@current@cmd in \@@enc@update. ASAJ. 277
1994-11-30 ltmath.dtx 1.0q
General: ASAJ: \DeclareMathOperator moved to AMSLaTeX. 481
1994-11-30 ltmiscen.dtx v1.0w
\@enddocument@kernel@warnings: (DPC) Do warnings even for \nofiles 462
\enddocument: (DPC) Use \@dofilelist 462
1994-11-30 ltoutenc.dtx 1.7a
General: Redefined \a for the new scheme. 225
1994-11-30 ltoutenc.dtx v1.6g
General: Removed new definitions of \patterns and \hyphenation, since encoding-specific commands now expand in the mouth. 225
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. 213
Always load the enc.def file, so that the default encoding for the commands will change. 255
Redefined \@changed@cmd to expand in the mouth. 217
Removed \@changed@x@mouth since \@changed@x now expands in the mouth. 217
Rewrote \@text@composite so it allows an empty argument, or an argument containing lots of commands. 219
1994-12-01 ltfinal.dtx v1.0p
General: Renamed lthyphen.* to hyphen.*. 797
1994-12-01 lthyphen.dtx v1.0g
General: Rename lthyphen.ltx/cfg to hyphen.ltx/cfg 795
1994-12-01 ltplain.dtx v1.1g
General: (DPC) More doc changes . . 14
1994-12-02 fontdef.dtx v2.2i
General: Commented out \ldots. ASAJ. 390
1994-12-02 ltfssini.dtx v2.2c
\copyright: \copyright is now in ltoutenc. ASAJ 387
1994-12-02 ltlists.dtx v1.0e
\@trivlist: RmS: Added check for looping 504
1994-12-02 ltoutenc.dtx 1.7b
General: Redefined \a properly. . . . 225
1994-12-02 ltoutenc.dtx v1.7b
General: Fixed a bug with \a. 213
1994-12-04 lthyphen.dtx v1.0h
General: Documentation edits for /1989 795
1994-12-05 ltoutenc.dtx v1.7c
General: Added braces to \textcircled. 213
1994-12-06 ltfssbas.dtx v2.1z
\DeclareFontEncoding: use \nfss@catcodes 272
\nfss@catcodes: Added tab char as well 282
1994-12-08 ltoutenc.dtx v1.7d
General: Added \null and \sh@ft to \b and \d. 213
1994-12-08 lttab.dtx v1.0k
\@array: Add \tabularnewline . . . 540
\tabularnewline: (DPC) Made it \relax 541
1994-12-09 ltbibl.dtx v1.1d
\bibliographystyle: (DPC) Allow use in preamble. 627
1994-12-10 ltfloat.dtx v1.1g
\@dblfloat: Old version reinstated temporarily 604
\@dblflset: Macro removed temporarily 604
Old version reinstated temporarily 604
\@setfps: Macro removed temporarily 605
\@xdblfloat: Macros reinserted temporarily 610

\@xfloat: Old version reinstated temporarily 605
Sanitisation added temporarily . . 605
General: Some temps reinserted temporarily 601
\fps@dbl: Macro removed temporarily 605
1994-12-10 ltfntcmd.dtx v3.3q
\@@math@egroup: Don't read arguments 421
\check@nocorr@: Use \space command for comparison 418
1994-12-10 ltfssdcl.dtx v2.1p
\document@select@group: Surround with braces (add fourth arg) . . . 345
\select@group: Surround with braces (add fourth arg) 343
1994-12-10 ltoutenc.dtx v1.7e
General: Added documentation for the OML encoding. 213
Replaced width with \@width and ditto height in vrules. 213
1994-12-14 ltoutenc.dtx v1.7f
General: Added braces to \copyright so it works unbraced in subscripts. 213
Added check for math mode in \@changed@cmd. 213
Commented out \textasciicircum, \textasciitilde, \textbackslash, \textbar, \textgreater, \texthyphenchar, \texthyphen and \textless to save memory. 213
1995-01-12 ltmath.dtx v1.2y classes
\@eqnnum: Added \normalcolor . . . 493
1995-03-03 ltoutenc.dtx 1.7g
General: Corrected an error in documentation referring to the tabular rather than the tabbing environment. 225
1995-04-02 ltfntcmd.dtx v3.3r
\@@math@egroup: Read them again to be able to add \relax. 421
1995-04-02 ltfssdcl.dtx v2.1q
\document@select@group: fix problem for pr/1275 345
\select@group: fix problem for pr/1275 343
\set@mathdelimiter: fix pr/1329 . . 362
1995-04-02 ltfssini.dtx v2.2d
\not@math@alphabet: add \noexpand to second part of message 385
1995-04-21 ltclass.dtx v1.0m
\DeclareOption*: Made long /1498 647
\endfilecontents: Close input check stream: latex/1487 662
1995-04-21 ltfinal.dtx v1.0q
General: Allow initial patch level 0 . 810
1995-04-21 ltoutenc.dtx v1.7h
General: Added \null \k latex/1274 213
1995-04-22 ltfiles.dtx v1.0p
\includeonly: Allow blanks in argument 201
1995-04-22 ltmiscen.dtx v1.0x
General: Removed extra def of \@gobble 460
1995-04-23 ltsect.dtx v1.0j
\addcontentsline: Use \contentsline internally. 597
1995-04-24 ltbibl.dtx v1.1e
\@citex: Add \mbox to undefined case: latex/1239. 626
1995-04-24 ltbibl.dtx v1.1f
\bibcite: Make \@onlypreamble /1388. 626
1995-04-24 ltcntrl.dtx v1.0d
\@for: Dont expand second argument with \edef: /1317 (DPC) 160
1995-04-24 ltoutput.dtx v1.1j
\fl@tracemessage: Do not add to kernel unless 'trace' specified . . . 780
1995-04-24 ltoutput.dtx v1.1l
\@begindvibox: Add \vbox latex/1392 734
\@writesetup: Reset \\ latex/1451 (DPC) 750
1995-04-24 ltpage.dtx v1.0f
\fussy: reset \emergencystretch latex/1344 632
1995-04-24 ltplain.dtx v1.1h
\newlanguage: Remove remaining \outer declarations. 17
1995-04-24 ltxref.dtx v1.1e
\newlabel: Make \@onlypreamble for /1388. 457
1995-04-25 ltdefns.dtx v1.2i
\@check@c: Make \long for latex/1346 78
\new@environment: Parse arguments slowly but safely /1507 76
1995-04-25 ltfiles.dtx v1.0q
\document: Removed execution of \every@size latex/1407 196
1995-04-25 ltsect.dtx v1.0k
\@dottedtocline: Added \hbox around dots. 599

1995-04-27 ltboxes.dtx v1.0s
\@frameb@x: Move \leavevmode for graphics/1512 519
\@iframebox: Move \leavevmode for graphics/1512 519
\@iirsbox: Move \leavevmode for graphics/1512 525
\@irsbox: Move \leavevmode for graphics/1512 525
\fbox: Move \leavevmode for graphics/1512 518
\raisebox: Move \leavevmode for graphics/1512 525
1995-04-27 ltfiles.dtx v1.0r
\document: Added \global to support groups in hook 197
1995-04-27 ltmiscen.dtx v1.0y
\enddocument: \@checkend moved after hook 461
1995-04-27 ltplain.dtx v1.1i
General: Move \hang and \textindent to latex209.def 29
1995-04-29 ltcntrl.dtx v1.0e
General: Moved init of \protect to ltdefns.dtx 160
Removed unused defs for \@setprotect and \@resetprotect 160
1995-04-29 ltdefns.dtx v1.2j
\protect: Init \protect here 82
1995-04-29 ltpar.dtx v1.1b
General: (TO) Comments clean-up. 171
1995-05-02 ltsect.dtx v1.0l
\@dottedtocline: Don't reset to \rmfamily 599
1995-05-03 ltsect.dtx v1.0m
General: TO: Promoted documentation to doc.sty standard 588
1995-05-06 ltsect.dtx 1.0n
\@seccntformat: Use \quad instead of \hskip 594
\@sect: Added \relax after \@seccntformat just in case . . . 592
1995-05-07 ltboxes.dtx v1.0t
General: Use \hb@xt@ 513
1995-05-07 ltdefns.dtx v1.2k
\hb@xt@: Macro added 71
1995-05-07 ltmath.dtx v1.0r
General: Use \hb@xt@ 481
1995-05-07 ltoutput.dtx v1.1m
General: Use \hb@xt@. 722
1995-05-07 ltpictur.dtx v1.0g
General: Use \hb@xt@ 552
1995-05-07 ltplain.dtx v1.1j
General: Use \hb@xt@ 14
1995-05-07 ltsect.dtx v1.0o
General: Use \hb@xt@ 588
1995-05-07 lttab.dtx v1.0l
General: Use \hb@xt@ 527
1995-05-08 ltbibl.dtx v1.1g
\@citex: Use \@firstofone 626
\bibitem: Removed unnecessary braces 626
\nocite: Use \@firstofone 627
1995-05-08 ltdefns.dtx v1.2k
\typein: Use \@firstofone 72
1995-05-08 ltdefns.dtx v1.2l
\typein: Remove unnecessary braces 72
Replace \def by \let 72
1995-05-08 ltfsstrc.dtx v2.3n
\ifnot@nil: Use \@firstofone . . . 324
1995-05-11 fontdef.dtx v2.2j
General: Updates to some plain macros 390
1995-05-12 ltclass.dtx v1.0n
\DeclareOption*: Use \toks@ to remove need to double hash /1557 647
1995-05-12 ltfloat.dtx v1.1h
\@footnotemark: Add \nobreak to allow hyphenation. latex/1605 . . 621
1995-05-12 ltpictur.dtx v1.0h
\pictur@: Macro added for latex/1355 554
1995-05-12 ltvers.dtx v1.0e
General: Add autoload docstrip guards 34
Check for format older than 1 year 34
1995-05-13 ltfsstrc.dtx v2.3o
General: Use single hash mark in \DeclareOption 313
1995-05-16 ltfloat.dtx v1.1i
\@makefnmark: Now use \textsuperscript. 618
\textsuperscript: Command added./pr1503 618
\thefootnote: Streamlined parts of code. 618
1995-05-17 ltboxes.dtx v1.0u
\@irsbox: Removed surplus braces . 525
1995-05-17 ltdefns.dtx v1.0o
\g@addto@macro: Make long for latex/1522 99
1995-05-17 ltlists.dtx v1.0g
\@item: Removed surplus braces . . . 508
\@nbitem: Removed surplus braces . 509
enumerate: Use \thr@@ and remove surplus braces 510
itemize: Use \thr@@ 511

1995-05-18 ltfloat.dtx v1.1j
\\@makefnmark: Added \\normalfont. 618
\\thempfootnote: Added \\itshape. . 618
1995-05-19 ltpictur.dtx v1.1a
General: Support autoloading feature 552
1995-05-20 ltcounts.dtx v1.1b
\\@definecounter: Streamlined code 260
\\@fnsymbol: Allowing both text and math 263
\\fnsymbol: Streamlined code 262
1995-05-20 ltcounts.dtx v1.1c
\\@definecounter: And do it right . 260
1995-05-20 ltfloat.dtx v1.1k
\\@makefnmark: Moved \\normalfont back and use \\@textsuperscript 618
Moved \\normalfont to \\textsuperscript 618
\\textsuperscript: Use \\normalfont. 618
1995-05-21 ltfssdcl.dtx v2.1t
\\DeclareMathRadical: Allow for undefined cs names 362
1995-05-21 ltlists.dtx v1.0f
General: Moved to doc.sty standard 496
1995-05-21 ltmath.dtx v1.0r
\\@sqrt: Use \\sqrtsign 490
General: Remove \\mathhexbox from this file 486
Update some plain macros 481
\\lefteqn: Use \\rlap 492
\\r@@t: Use \\sqrtsign instead of \\sqrt 483
1995-05-21 ltoutenc.dtx v1.7h
\\@inmathwarn: Added several \\@onlypreamble 217
1995-05-21 ltoutenc.dtx v1.7j
General: Updated some plain macros 229
1995-05-21 ltplain.dtx v1.1j
General: Moved some code to other files 14
1995-05-22 ltplain.dtx v1.1k
General: Definitions of \\footins and \\footnoterule moved to ltfloat. . 31
1995-05-22 lttab.dtx v1.1a
General: Support autoloading feature 527
1995-05-23 ltfssini.dtx v2.2e
\\newfont: Font assignment made local again. 385
1995-05-24 ltdefns.dtx v1.1l
\\newif: (DPC) New implementation . 77
1995-05-24 ltdefns.dtx v1.2m
\\typein: (DPC) New implementation 72
1995-05-24 ltfloat.dtx v1.1l
\\@textsuperscript: Command added. 618
General: Moved definition of \\footins and \\footnoterule from ltplain. 618
\\textsuperscript: Use \\@textsuperscript 618
1995-05-24 ltfssbas.dtx v3.0a
General: (DPC) Make file from previous file, fam.dtx 1995/05/20 v2.2d 268
\\mathgroup: (DPC) No need to redefine \\newfam as not outer . . 268
1995-05-24 ltfsscmp.dtx v3.0a
General: (DPC) Make file from previous file, fam.dtx 1995/05/20 v2.2d 335
1995-05-24 ltfssdcl.dtx v3.0a
General: (DPC) Make file from previous file, latint.dtx 1995/05/21 v2.1t 340
1995-05-24 ltfssini.dtx v3.0a
General: (DPC) Make file from previous file, lfonts.dtx 1995/05/23 v2.2e 367
\\cal: (DPC) Remove definition . . . 389
\\mit: (DPC) Remove definition . . . 389
1995-05-24 ltfsstrc.dtx v3.0a
General: (DPC) Make file from previous file, tracefnt 1995/05/16 v2.3o 312
1995-05-24 ltfsstrc.dtx v3.0b
General: (DPC) Fix \\ProvidesFile usage 312
1995-05-25 ltclass.dtx v1.0p
\\endfilecontents: Delete \\filec@ntents after preamble . . 662
1995-05-25 ltfilehook.dtx v1.0t
\\unqu@tefilef@und: (CAR) added \\long 689
1995-05-25 ltfiles.dtx v1.0s
\\document: Added check for \\topskip zero 197
1995-05-25 ltfiles.dtx v1.0t
\\@iffileonpath: (CAR) added \\long 207
\\document: Corrected typo 197
\\IfFileExists@: (CAR) added \\long 206
\\nofiles: (CAR) added \\long 200
\\protected@write: (CAR) added \\long 201
1995-05-25 ltfloat.dtx v1.1m
\\@savemarbox: (CAR) Resettings moved to hook 614
\\@xfloat: (CAR) Resettings moved to hook 606

1995-05-25 ltlists.dtx v1.0i
\endtrivlist: Macros moved from ltspace.dtx 505
1995-05-25 ltmath.dtx v1.3c classes
\@eqnnum: replace \reset@font\rmfamily with \normalfont (PR 1578) 493
1995-05-25 ltspace.dtx v1.2f
\@vbsphack: (CAR) not used so 'removed'. 183
\@vspacer: (CAR) \@restorepar added to avoid possible infinite tail recursion caused by a typo in the argument. 186
(CAR) macros modified to be more efficient 186
General: Macros moved to ltlists.dtx 173
1995-05-26 ltdefns.dtx v1.2n
\@gobblefour: (CAR) Added \longs 79
1995-05-26 ltmath.dtx v1.0s
\@eqnnum: Removed \rmfamily (PR 1578), replaced \reset@font with \normalfont 490
1995-05-26 ltpage.dtx v1.0g
\ps@plain: removed \rmfamily (PR 1578) 630
1995-05-27 ltfssbas.dtx v3.0b
\mathgroup: (FMi) But a need to define \new@mathgroup 268
1995-06-05 fontdef.dtx v2.2k
General: Moved math commands from ltoutenc.dtx. 408
1995-06-05 ltfinal.dtx v1.0r
General: Added \MakeUppercase and \MakeLowercase. 797
1995-06-05 ltoutenc.dtx v1.7k
\@inmathwarn: Removed \protected@cmd and replaced with explicit \noexpand. 217
General: Allowed \ProvideTextCommandDefault after the preamble. 219
Commented out \textless and \textgreater. 227
Moved math commands to fontdef.dtx. 229
Save some tokens in \textvisiblespace and \textunderscore. 227
1995-06-06 ltfinal.dtx v1.0s
General: Made \MakeUppercase and \MakeLowercase brace their argument. 797
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. 220
1995-06-11 ltspace.dtx v1.2g
\restorecr: (CAR) \relax added to stop silent eating of *. 191
1995-06-13 ltfinal.dtx v1.0t
General: Add patch level string more carefully 810
Call \errorstopmode 811
1995-06-13 ltpictur.dtx v1.1b
General: Use \ProvidesFile in autoload 552
1995-06-14 lttab.dtx v1.1b
General: Use \ProvidesFile in autoload 527
1995-06-15 ltfssbas.dtx v3.0c
General: (DPC) minor documentation changes 268
1995-06-15 ltfsscmp.dtx v3.0b
General: (DPC) minor documentation edits 335
1995-06-15 ltfssdcl.dtx v3.0b
General: (DPC) minor documentation changes 340
1995-06-19 ltbibl.dtx v1.1h
\bibcite: Call \@newl@bel so repeated keys produce better warning. 626
1995-06-19 ltclass.dtx v1.0q
\documentclass: Dont redefine \usepackage in compat mode for /1634 650
1995-06-19 ltxref.dtx v1.1e
\newlabel: Use \@newl@bel to share code with \bibcite 457
1995-06-28 ltfssini.dtx v3.0b
General: (DPC) Fix documentation typos 367
1995-06-28 ltmath.dtx v1.0t
General: minor doc edits 481
1995-07-02 ltplain.dtx v1.1n
General: Removed surplus 'by' and '=' in various places 14
\offinterlineskip: Replaced 1000 by \@m 28
\showoutput: Use \showoverfull to save space 31
\tracingall: Use \showoutput to save space 31

1995-07-03 ltdefns.dtx v1.2o
\set@typeset@protect: Use \@typeset@protect for init 82
1995-07-03 ltfntcmd.dtx v3.3s
\t@st@ic: Use clean interface for jump 420
1995-07-05 ltfntcmd.dtx v3.3s
\t@st@ic: Renamed from \test@next 420
1995-07-05 ltspace.dtx v1.2h
\@gnewline: Use \break 179
\@no@pgbk: Macro replaces \@pgbk and \@nopgbk 177
\nopagebreak: Reimplemented both using \@no@pgbk 176
1995-07-09 ltcntrl.dtx v1.0f
\@iforloop: Reimplemented using Kabelschacht method 160
\@iwhiledim: Reimplemented using Kabelschacht method 158
\@iwhilenum: Reimplemented using Kabelschacht method 158
\@iwhilesw: Reimplemented using Kabelschacht method 158
\@tfor: Reimplemented using Kabelschacht method 160
1995-07-09 ltlists.dtx v1.0j
enumerate: Use \expandafter 510
itemize: Use \expandafter 511
1995-07-12 ltpictur.dtx v1.1d
General: allow 2e commands in 209 mode. latex/1737 552
1995-07-13 ltdefns.dtx v1.0p
General: Updates to documentation . 70
1995-07-13 ltfiles.dtx v1.0u
General: Updates to docu 193
1995-07-13 ltfssbas.dtx v3.0d
\@@defaultsubs: macro added 287
\@defaultsubs: macro added 287
General: minor documentation changes 268
\wrong@fontshape: Change a macro not a switch to flag default font substitutions 286
1995-07-13 ltmiscen.dtx v1.0z
\@centercr: Use \nobreak 472
\@enddocument@kernel@warnings: Use \@defaultsubs instead of switch 463
\@writefile: Added missing percent and use \relax in the THEN case 466
\@xobeysp: Use \nobreak 474
General: Improve Documentation . . 460
\enddocument: Set \@setckpt to \@gobbletwo instead of defining it by hand 461
Shorten redefinition of \bibcite and \newlabel 462
1995-07-14 ltbibl.dtx v1.1i
\bibcite: Remove \@onlypreamble so still defined in new \enddocument 626
1995-07-14 ltxref.dtx v1.1g
\newlabel: Remove \@onlypreamble so still defined in new \enddocument 457
1995-07-19 ltfssini.dtx v3.0d
General: (DPC) TeX2 support 388
1995-07-20 ltboxes.dtx v1.0v
\@isavebox: Use \sbox 517
\@isavepicbox: Use \sbox 517
1995-07-21 ltoutput.dtx v1.1o
\@writesetup: Command added . . . 749
New, experimental, versions: need in-lining 749
1995-08-09 ltmath.dtx v1.0u
General: Added code for class options leqno and fleqn 493
1995-08-11 ltlength.dtx v1.1b
General: Doc typos fixed for latex/753 266
1995-08-16 ltcntrl.dtx v1.0g
\@break@tfor: Made long 160
\@forloop: Made defs long 160
\@fornoop: Made defs long 160
\@iforloop: Made defs long 160
\@iwhiledim: Made defs long 158
Removed \@whilenoop 158
\@iwhilenum: Made defs long 158
Removed \@whilenoop 158
\@iwhilesw: Removed \@whileswnoop 158
\@tfor: Made defs long 160
1995-08-16 ltfiles.dtx v1.0v
\document: set \@maxdepth 197
set \do globally 197
set \topskip globally 197
1995-08-24 ltfssbas.dtx v3.0f
General: Added autoload code 268
1995-08-24 ltfsstrc.dtx v3.0c
General: Macro \gobble@font@spec removed 324
\tryis@simple: 331
1995-08-25 ltoutput.dtx v1.1p
General: Support autoloading feature (FMi). 722
1995-09-01 lterror.dtx v1.2i
General: Add autoload support . . . 161
1995-09-01 ltplain.dtx v1.1m
\empty: Use \let to save space 27
\I: Use \let to save space 26

1995-09-14 ltplain.dtx v1.1o
General: Moved \multispan to lttab.dtx 14
1995-09-14 lttab.dtx v1.1c
\cline: (DPC) New implementation 550
1995-09-15 ltfssini.dtx v3.0e
General: (DPC) Modify TeX2 message 388
1995-09-19 ltmiscen.dtx v1.1a
\verb: Put \@noligs after \verbatim@font where it belongs. 479
1995-10-01 ltfiles.dtx LaTeX2e
\@addtofilelist: Macro added . . . 211
1995-10-02 ltdefns.dtx v1.2q
\@ifnch: Use \@let@token for internal/924, save \reserved@e . . 95
\@ifnextchar: Use \@let@token . . . 95
\@protected@testopt: Macro added . 75
\@testopt: Macro added 74
\@xargdef: New implementation, using \@test@opt 74
1995-10-03 fontdef.dtx v2.2l
General: \@@sqrt from patch file for /1701 390
1995-10-03 ltdefns.dtx v1.2r
\typein: Add missing \@typein for /1710 (from patch file) 72
1995-10-03 ltpictur.dtx v1.1e
General: New autoload code 552
1995-10-04 ltfssbas.dtx v3.0g
General: Modify autoload code 268
1995-10-04 ltfsstrc.dtx v3.0d
General: (DPC) Modify autoload code 312
1995-10-04 lttab.dtx v1.1d
General: Modify autoload support . 527
1995-10-06 ltfiles.dtx v1.0w
\@missingfileerror: Autoload error 209
1995-10-09 lterror.dtx v1.2j
General: Modify autoload support . 161
1995-10-09 ltoutenc.dtx v1.7m
\@inmathwarn: Autoload error 218
1995-10-10 ltfssbas.dtx v3.0h
\showhyphens: Use \normalfont and make colour safe, and autoloadable 291
1995-10-10 ltfssdcl.dtx v3.0c
\non@alpherr: (DPC) autoload error message 344
1995-10-10 ltplain.dtx v1.1r
General: Autoload tracing code 14
1995-10-10 ltthm.dtx v1.0f
General: Make \newtheorem 'only preamble' 584
1995-10-11 ltoutput.dtx v1.1r
\clearpage: Added a check so that it does not lose the argument of \twocolumn[...] 735
1995-10-16 ltbibl.dtx v1.1j
\cite: (DPC) Make robust 626
1995-10-16 ltboxes.dtx v1.0w
General: Clarify makebox description 513
1995-10-16 ltdefns.dtx v1.2u
\@ifstar: (DPC) New implementation, for /1910 96
\new@command: (DPC) Use \@testopt /1911 74
\new@environment: (DPC) Use \@testopt /1911 76
\typein: (DPC) Use \@testopt /1911 72
1995-10-16 ltfssini.dtx v3.0f
\reset@font: Added \relax after \usefont, as the latter eats up spaces. 387
1995-10-16 ltmath.dtx v1.0y
\@yeqncr: (DPC) Use \@testopt /1911 491
\sqrt: (DPC) Make robust /1808 . . 490
1995-10-16 ltspace.dtx v1.2j
\nolinebreak: (DPC) Use \@testopt /1911 176
\nopagebreak: (DPC) Use \@testopt /1911 176
1995-10-16 ltthm.dtx v1.0g
General: Revert to previous \newtheorem behaviour 584
1995-10-17 ltclass.dtx v1.0r
\@providesfile: Delay definition of \ProvidesFile till ltfinal 646
\ProcessOptions*: Reset \CurrentOption for graphics/1873 649
1995-10-17 ltdirchk.dtx v1.0l
General: Modify initex version of \ProvidesFile 4
1995-10-17 ltfinal.dtx v1.0v
\@providesfile: reset macro 811
\reserved@b: reset here after the \input above 810
1995-10-17 ltplain.dtx v1.1s
\eject: Move \superejected to compat file 29
1995-10-17 lttab.dtx v1.1e
\@cline: (DPC) Use \@multicnt . . 550
\@multispan: (DPC) Macro added. 550
1995-10-19 ltfinal.dtx v1.0w
\@filelist: Move after \reserved@a setting:-) 811

1995-10-20 ltbibl.dtx v1.1k
\@citex: Removed refundefined flag 626
\nocite: Removed refundefined flag 627
1995-10-20 ltclass.dtx v1.0s
\@begindocumenthook: Make setting conditional, for autoload version 660
1995-10-20 ltfssbas.dtx v3.0i
General: (DPC) Modify autoload code, change \undefined 268
1995-10-20 ltfsstrc.dtx v3.0e
General: (DPC) Modify autoload code 312
1995-10-22 ltfssbas.dtx v3.0j
General: (RmS) New size function macro \genb@sfcnt needs to be disabled at \document. 268
1995-10-22 ltfsstrc.dtx v3.0f
General: Added 'genb' and 'sgenb' size functions to support new DC font naming scheme. 312
1995-10-23 lttab.dtx v1.1f
\@settab: (CAR)Ensure that \@hightab increases by at most one 534
\@startline: (CAR)Ensure that \@nxttabmar is never larger than \@hightab 532
\poptabs: (CAR)Ensure that \@curtab is never larger than \@hightab 535
\tabbing: (CAR)Make \@hightab consistently a local variable 534
1995-10-24 ltfiles.dtx v1.1a
\document: Removed multiplelabels switch 196
Removed refundefined switch . . . 197
1995-10-24 ltfssbas.dtx v3.0k
\@@defaultsubs: macro removed . . 287
\wrong@fontshape: Make this code inline since it happens only here 286
1995-10-24 ltmiscen.dtx v1.1b
\@enddocument@kernel@warnings: Changed logic for producing warning messages and removed switch 463
Use \@refundefined instead of switch 463
1995-10-24 ltxref.dtx v1.1h
\@multiplelabels: Switch for multiplelabels removed 457
\@newl@bel: Switch for multiplelabels replaced by inline code 457
\@refundefined: Switch for refundefined replaced 456
\@setref: Switch for refundefined renamed 457
\if@multiplelabels: Macro removed 457
1995-10-25 ltalloc.dtx v1.1b
General: General doc improvements 155
1995-10-25 ltfloat.dtx v1.1n
\@endfloatbox: (CAR) macro added: to unify code for double and single versions 610
\end@dblfloat: (CAR) unify code for double and single versions 609
\end@float: (CAR) unify code for double and single versions 608
1995-10-25 ltidxglo.dtx v1.1d
General: Doc cleanup 622
1995-10-25 ltsect.dtx v1.0q
\subparagraphmark: Use \let not \def to save space. 596
1995-10-27 ltpictur.dtx v1.1f
General: Move initialisation to kernel from autoload file 579
1995-10-31 ltboxes.dtx v1.0x
\@finalstrut: Add \nobreak in horiz mode to allow hyphenation. internal/1931 526
1995-11-01 fontdef.dtx v2.2m
General: add \nfss@catcodes for internal/1932 393
1995-11-01 ltdirchk.dtx v1.0n
General: Initialise \@addtofilelist to \@gobble 4
1995-11-01 ltfinal.dtx v1.0x
General: (DPC) Switch meaning of \@addtofilelist for cfg files . . . 802
1995-11-01 ltfssbas.dtx v3.0m
\DeclareFontShape@: (DPC) Test for \relax not \undefined, internal/1933 269
1995-11-01 ltfssini.dtx v3.0g
General: (DPC) Switch meaning of \@addtofilelist for cfg files . . . 388
1995-11-02 ltfssbas.dtx v3.0n
\wrong@fontshape: (DPC) Remove extra space with \string for latex/1676 286
1995-11-02 ltoutenc.dtx v1.7n
General: Changed internal name \a to \@tabacckludge to protect against redefinition by malicious users. . 225
1995-11-07 ltlists.dtx v1.0k
\@doendpe: Enclosed \setbox0 assignment by a group so that it leaves the contents of box 0 intact. 506

1995-11-07 ltoutenc.dtx v1.7o
General: Added \leavevmode at start of \c, otherwise the output routine might be invoked within the macro. 229
Changed \char32 to \@xxxii (two tokens less). 230
Replaced octal number 27 by decimal number 23 to protect against the quote character being active. 230
Replaced some 0's by \z@ (faster). 230
1995-11-10 ltoutput.dtx v1.1s
\@shipoutsetup: Command removed 749
\@writesetup: Command removed . 749
In-lined 749
1995-11-14 ltclass.dtx v1.0t
\@@unprocessedoptions: Allow empty option 661
\@loadwithoptions: macro added . 651
\LoadClassWithOptions: macro added 651
\RequirePackageWithOptions: macro added 651
1995-11-17 ltfssbas.dtx v3.0m
\@wrong@font@char: (DPC) Macro added. latex/1676 287
\define@newfont: Redefine \typeout latex/1676 280
\wrong@fontshape: Support \@wrong@font@char latex/1676 . 286
1995-11-17 ltoutenc.dtx v1.7p
\UseTextSymbol: Support \@wrong@font@char latex/1676 . 222
1995-11-18 ltoutenc.dtx v1.7q
\UseTextSymbol: Modify message slightly 222
1995-11-21 fontdef.dtx v2.2n
General: Incorporate changed figures, as in plain.tex 406
1995-11-27 ltfssbas.dtx v3.0n
\nfss@catcodes: Reset hash, for definitions in fd files 282
1995-11-28 ltfloat.dtx v1.1n
General: documentation fixes 601
1995-11-28 ltfsstrc.dtx v3.0g
General: documentation fixes 312
1995-11-28 ltoutenc.dtx v1.7r
General: Added math mode checks to text commands. 217
doc fixes 213
Renamed \@changed@x@err to \TextSymbolUnavailable. 217
1995-11-29 ltoutenc.dtx v1.7t
General: Added \textasciicircum, \textasciitilde, \textbackslash, \textbar, \textgreater and \textless. . . 233
Added \textasciicircum, \textasciitilde, \textregistered and \texttrademark. 227
Added \textbackslash and \textbar. 226, 237
Added \textless and \textgreater. 227, 238
1995-12-01 ltoutenc.dtx v1.7u
General: Made \SS a Default, rather than having the default point to the OT1 definition. 227
1995-12-04 ltspace.dtx v1.2k
\nobreakspace: (Macro added 189
1995-12-04 ltspace.dtx v1.2l
\@xobeysp: (braces added to definition of tilde 189
1995-12-04 preload.dtx v2.4e
General: Ulrik Vieth. added 12pt OMS and OML preloads /1989 . 413
1995-12-05 ltdefns.dtx 1.2w
\@unexpandable@protect: Removed \unexpandable@noexpand as never used. internal/1733 80
1995-12-05 ltfiles.dtx v1.1c
\document: \ignorespaces added for latex/1933 197
1995-12-05 ltfloat.dtx v1.1n
\@textsuperscript: Use \ensuremath for latex/1984. 618
1995-12-05 ltoutenc.dtx v1.7v
\@inmathwarn: Changed \TextSymbolUnavailable text . . 218
1995-12-06 ltfssbas.dtx v3.00
\nfss@catcodes: Reset hat, for typeouts etc in fd files 282
1995-12-07 ltbibl.dtx v1.1l
\@citex: Restored name of \G@refundefinedtrue 626
1995-12-07 ltfloat.dtx v1.1m
\@textsuperscript: Move \m@th out of the \ensuremath for latex/1984. 618
1995-12-07 ltxref.dtx v1.1i
\@setref: Switch for refundefined restored 457
\G@refundefinedtrue: Renamed (back) from \G@refundefined . . 456

1995-12-11 ltoutenc.dtx v1.7w
General: Modified \copyright 227
1995-12-13 ltdefns.dtx 1.2x
\-: Documentation changed. 97
1996-01-10 ltfiles.dtx v1.1d
\@iffileonpath: Change argument handling to not require doubled hash. latex/2024 207
1996-01-20 ltidxglo.dtx v1.1e
\makeglossary: Make no-op after use pr/2048 623
\makeindex: Make no-op after use pr/2048 623
1996-01-20 ltspace.dtx v1.2m
\vspace: Made robust 186
1996-03-25 ltmath.dtx v1.1a
\@ensuredmath: Macro added for amslatex/2104 493
\ensuremath: Reimplement for amslatex/2104 492
1996-04-18 ltpage.dtx v1.0i
General: Improve documentation . . 629
1996-04-22 ltmiscen.dtx v1.1c
General: Improve Documentation . . 460
1996-04-22 ltspace.dtx v1.2n
General: Documentation Improvements 173
1996-04-22 lttab.dtx v1.1g
\@tabclassz: (DPC) Extra \hskip keeps tabcolsep in empty columns internal/2122 547
1996-04-23 ltcounts.dtx v1.1d
General: Documentation improvements 258
1996-04-24 ltfiles.dtx v1.1e
\document: (DPC) Reset \AtBeginDocument eg for latex/1297 197
1996-05-08 ltfsstrc.dtx v3.0h
\math@egroup: Use \bgroup instead of \begingroup to match a kernel change made in 1994!! 323
1996-05-09 ltfntcmd.dtx v3.3t
\check@icr: Default definitions added 418
1996-05-17 fontdef.dtx v2.2o
General: \@@sqrt removed, at last 390, 406
1996-05-17 ltfiles.dtx v1.1f
\nofiles: added \write to \protected@write for latex/2146 200
1996-05-18 ltoutenc.dtx v1.7x
General: Produce error if encoding not found. pr/2054 255
1996-05-21 ltoutenc.dtx v1.7y
General: Corrected error message (CAR) 255
1996-05-21 ltsect.dtx v1.0s
\@sect: (DPC) Added extra braces for internal/2148 592
(DPC) Moved brace to allow commands like \MakeUppercase in 6th argument. Changed \par to \endgraf to allow non-long commands. internal/2148 592
\@ssect: (DPC) Added extra braces for internal/2148 595
(DPC) Moved brace to allow commands like \MakeUppercase in 4th argument. Changed \par to \endgraf to allow non-long commands. internal/2148 595
1996-05-23 ltoutenc.dtx v1.7z
\@strip@args: \expandafter added to match other changes for latex/2133 222
\add@accent: macro added. latex/2133 219
\DeclareTextAccent: Reimplemented using \add@accent to save space latex/2133 219
\DeclareTextCompositeCommand: Modified to cope with new \add@accent command: required removal of check for one argument-command 220
1996-05-24 ltoutput.dtx v1.1t
\@specialoutput: Check that \@colroom is less than \vsize, indicating that a float has been added 740
Cut-off point changed to 1.5\baselineskip 740
\@topnewpage: Cut-off point changed to 2.5\baselineskip 739
1996-05-25 ltoutput.dtx v1.1u
\@specialoutput: Correct the above check 740
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. 475
General: Move setting of verbatim font and \@noligs. 460
\verb: Put setting of verbatim font after \dospecials so that \dospecials cannot reset the

category code of characters handled by \@noligs. 479
1996-06-10 ltboxes.dtx v1.0y
\@parboxto: (DPC) Changed \endgraf to \@@par 520
1996-06-10 ltsect.dtx v1.0t
\@sect: (DPC) Changed \endgraf to \@@par 592
\@ssect: (DPC) Changed \endgraf to \@@par 595
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 547
1996-06-22 ltspace.dtx v1.2o
General: Documentation of problems added 173
1996-07-10 ltfinal.dtx v1.0y
\toks: Free up memory from scratch registers /2213 810
1996-07-19 ltoutenc.dtx v1.8a
\@strip@args: Use char 0 not @ as carrier for \lowercase /2197 . . . 222
1996-07-26 ltboxes.dtx v1.0z
\if@minipage: put \global into definition 522
1996-07-26 ltclass.dtx v1.0u
\@classoptionslist: made only preamble 637
\@unusedoptionlist: made only preamble 638
1996-07-26 ltdefns.dtx v1.2y
\@reargdef: third arg picked up by \@yargdef 75
\renew@command: use \noexpand instead of \string 76
use \relax in place of empty arg . 76
\renew@environment: use \relax in place of empty arg 77
1996-07-26 ltfloat.dtx v1.1n
\@endfloatbox: remove unecessary \global before \@minipage... . 610
\@savemarbox: remove unecessary \global before \@minipage... . 614
\@setminipage: remove unecessary \global before \@minipage... . 608
\@setnobreak: remove unecessary \global before \@nobreak... . . 608
1996-07-26 ltfssbas.dtx v3.0p
\@DeclareMathSizes: use faster \if test 275
\nfss@catcodes: omit \relax as not needed 282
1996-07-26 ltfssdcl.dtx v3.0e
\init@restore@version: Removed \ifrestore@version switch and replaced by \init@restore@version 344
1996-07-26 ltfsstrc.dtx v3.0i
\init@restore@glb@settings: macro added replacing \if@inmath switch 321
1996-07-26 ltlists.dtx v1.0l
\@item: Remove unecessary \global before \@minipage... 507
Remove unecessary \global before \@nobreak... 508
1996-07-26 ltmath.dtx v1.1b
General: Removed \global before \@ignoretrue in various places. . 481
1996-07-26 ltmiscen.dtx v1.1e
\@ignorefalse: put \global into definition 461
\begin: remove \global before \@ignore... 467
\end: remove \global before \@ignore... 470
\ignorespacesafterend: user level macro added 461
1996-07-26 ltoutput.dtx v1.1v
\@testfp: remove \global before \@test... 783
\@xtryfc: remove \global before \@test... 757
\@ztryfc: remove \global before \@test... 758
General: put \global into definition 732
remove \global before \@test... 732
\clearpage: add number of missing percents 735
1996-07-26 ltplain.dtx v1.1t
\sh@ft: replace \dimen\z@ by \dimen@ 30
1996-07-26 ltsect.dtx v1.0u
\@starttoc: removed \global before \@nobreak... 597
\@xsect: Removed \global before \@nobreak... 593
1996-07-26 ltspace.dtx v1.2p
\if@nobreak: put \global inside definition 179
1996-07-27 ltfssbas.dtx v3.0q
General: \if@inmath switch removed 279
1996-07-27 ltspace.dtx v1.2q
General: Further documentation of problems 173

1996-07-27 ltspace.dtx v1.2r
General: Correct documentation of problems 173
1996-08-02 ltfloat.dtx v1.1o
\@xympar: Remove \global before \@ignore... 614
1996-08-02 ltsect.dtx v1.0v
\@afterheading: Removed \global before \@nobreak... 595
1996-08-02 ltspace.dtx v1.2s
\@Esphack: Remove \global before \@ignore... 182
1996-08-25 ltfssbas.dtx v3.0r
\nfss@catcodes: Reset the acute, grave and double quote chars as well 282
1996-09-21 ltoutput.dtx v1.1w
\@writesetup: Added \@parboxrestore and made consequent deletions: wait for the howls of protest 749
1996-09-25 ltdirchk.dtx v1.0t
General: Move ltxcheck to separate file 13
1996-09-28 ltmiscen.dtx v1.1f
\@xobeysp: Moved to ltspace.dtx . . 474
1996-09-28 ltspace.dtx v1.2t
\@xobeysp: Moved from ltmiscen.dtx and redefined to use \nobreakspace 189
1996-09-29 ltfiles.dtx v1.1g
\document: Added disabling of \@nodocument 197
1996-09-29 ltoutput.dtx v1.1x
\newpage: Checks for noskipsec and inlabel added 736
1996-09-29 ltsect.dtx 1.0w
\@noskipsectrue: Added documentation 589
1996-09-30 ltoutput.dtx v1.1y
\newpage: Checks for noskipsec and inlabel removed pending further tests 736
1996-10-04 ltclass.dtx v1.0v
\RequirePackageWithOptions: Reset \@unprocessedoptions for /2269 651
1996-10-05 ltfiles.dtx v1.1h
\@clubpenalty: Added setting its value 195
1996-10-08 ltfntcmd.dtx v3.3u
\DeclareTextFontCommand: Removed \check@icr when in vmode since it causes various errors (see pr/2157) 416
1996-10-21 lttab.dtx v1.1i
\@array: Use \set@typeset@protect 540
General: Moved the code associated with \@mkpream into the group provided by the box, for robustness (latex/2183) 539
\multicolumn: Make \multicolumn long (latex/2180) 542
\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) 534
1996-10-23 ltlists.dtx v1.0m
\@item: \@nobreak... moved into the \everypar and not executed unconditionally, see above 508
\kern... changed to \setbox... 508
Added setting of \clubpenalty and set \@nobreakfalse only when necessary 508
1996-10-23 ltsect.dtx v1.0x
\@xsect: Replaced \hskip... with \setbox... as used in \@afterheading 593
1996-10-24 ltboxes.dtx v1.1a
\@arrayparboxrestore: Added local settings of flags: dangerous! 521
\@iiiminipage: Use it or lose it (@setminpage): Frank will want to lose it 523
1996-10-24 ltfloat.dtx v1.1p
\@floatboxreset: Added local settings of flags: dangerous! 608
\@marginparreset: Added local settings of flags: dangerous! 614
\@xfloat: Added \@nodocument to trap floats in the preamble 605
1996-10-24 ltoutput.dtx v1.1z
\@addtocurcol: Added \nobreak, etc as appropriate 763, 767
\@specialoutput: Added \nobreak as appropriate 742
\@topnewpage: Added \@nodocument to trap \twocolumn in the preamble 738
\newpage: Better checks for noskipsec and inlabel added, plus nobreak . 736
1996-10-25 ltlists.dtx v1.0n
\endtrivlist: Change \indent to \leavevmode 505
Reset flags explicitly 505
1996-10-25 ltoutput.dtx v1.2a
\newpage: Reset all flags explicitly . 736

1996-10-26 ltlists.dtx v1.0o
\endtrivlist: Correct typo 505
1996-10-27 ltoutenc.dtx v1.8c
\@strip@args: Removed macro . . . 220
General: Added \r A 230
Added \textasteriskcentered . . 226, 237
Corrected syntax descriptions . . . 214
Removed \aa and \AA . 226, 230, 233
1996-10-28 ltplain.dtx v1.1u
General: (CAR) More doc changes . . 14
\dotfill: Removed math mode 31
1996-10-29 ltplain.dtx v1.1v
\dotfill: Got arithmetic correct (CAR) 31
1996-10-29 ltspace.dtx v1.2u
\@gnewline: Added macro 179
\@no@lnbk: Macro replaces \@lnbk and \@nolnbk 177
\\: Corrected and rationalised code 177
\nolinebreak: Reimplemented both using \@no@lnbk 176
1996-10-31 ltfinal.dtx v1.0z
General: Added extra \lcode, hoping it does no harm in T1 (pr/1969) 802, 808
1996-10-31 ltlists.dtx v1.0p
\@trivlist: Added check for missing item in outer list 504
1996-10-31 ltsect.dtx v1.0y
General: Corrected and tidied documentation; removed long lines 588
1996-11-03 ltplain.dtx v1.1w
\dotfill: Saved tokens by using \hb@xt@ 31
1996-11-04 lterror.dtx v1.2m
\@nodocument: Always define \@nodocument in kernel, so that it can be cleared by \document. . . . 167
1996-11-04 ltlists.dtx v1.0q
\@trivlist: Moved check for missing item: only checked when not inlabel flag is false 504
1996-11-05 ltfiles.dtx v1.1i
\nofiles: Standard \if@nobreak test added 200
1996-11-09 ltmath.dtx v1.1c
\@ensuredmath: Made long, as it was before. /2104 493
1996-11-18 ltfssbas.dtx v3.0s
\define@newfont: (DPC) lowercase fd file names. internal/1044 281
1996-11-18 ltoutenc.dtx v1.8d
General: (DPC) lowercase external file names. internal/1044 255
1996-11-20 fontdef.dtx v2.2p
General: lowercase fd and enc.def file names /1044 390
1996-11-20 ltvers.dtx v1.0f
General: Check for old format modified /2319 34
1996-11-23 ltoutenc.dtx v1.8e
General: Corrected description 214
Extended description 215
1996-11-28 ltvers.dtx v1.0g
General: Check for old format modified /2319 34
1996-12-06 ltdirchk.dtx v1.0u
\IfFileExists: *** removed from various messages for GNU Make. internal/2338 10
1996-12-06 ltfloat.dtx v1.1r
\@caption: Call \@setminpage if needed. latex/2318 604
1996-12-06 ltfssini.dtx v3.0h
General: (DPC) Remove *** from messages internal/2338 388
1996-12-17 ltdefns.dtx v1.0w
\g@addto@macro: Use \begingroup to save making a mathord 99
1996-12-20 ltsect.dtx v1.0z
\@dottedtocline: Added \nobreak for latex/2343 599
1997-01-08 fontdef.dtx v2.2q
General: Use \DeclareMathDelimiter to set delimiter codes 399
\mathparagraph: Define using \DeclareMathSymbol 408
1997-01-08 ltfiles.dtx v1.1j
\@include: reset \deadcycles latex/2365 204
1997-01-08 ltmath.dtx v1.1d
\root: (DPC) Remove spurious space tokens from plain TEX definition /2359 483
1997-02-05 ltdefns.dtx v1.0x
\g@addto@macro: missing percent /2402 99
1997-02-21 ltlists.dtx v1.0r
\@item: \ifvoid check added for \noindent. latex/2414 508
1997-03-21 ltcounts.dtx v1.1e
\fnsymbol: Use \mathsection and \mathparagraph. latex/2445 . . . 262

1997-04-14 ltfiles.dtx v1.1k
\document: Set the document space factor defaults. latex/2404 196
\normalsfcodes: Macro added (from patch file) latex/2404 200
1997-04-14 ltoutput.dtx v1.2b
\@writesetup: Call \normalsfcodes (from patch file) latex/2404 751
Move \label and \index (from patch file) 751
1997-04-24 ltbibl.dtx v1.1m
\@citex: \@empty to avoid primitive error on empty cite keys. latex/2432 626
1997-04-30 ltoutenc.dtx v1.9a
General: Changed \textsc to \scshape 227
Introduced \textcopyright and modified \copyright 227
Introduced \textcopyright and modify \copyright 228
Modified \textunderscore, removing \mathunderscore 227
Modified \underscore, removing \mathunderscore 228
1997-04-30 ltoutenc.dtx v1.9b
General: Added \leavevmode to \textunderscore 227
1997-05-04 ltoutenc.dtx v1.9c
General: Added 'hex index tabs' . . . 234
Added TS1 encoding v2.2.beta . . 240
1997-05-07 ltoutenc.dtx v1.9d
General: Added \leavevmode to \textcompwordmark 227
1997-05-07 ltspace.dtx v1.2v
\newline: Made completely robust. 178
1997-05-29 ltfsstrc.dtx v3.0j
General: Replaced \\ by \MessageBreak, as suggested by Donald Arseneau. 314
1997-05-29 ltlogos.dtx v1.1f
\LaTeXe: Added \m@th so that the LaTeX 2ε logo works with non-zero values of \mathsurround. 192
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 601
1997-06-16 ltfntcmd.dtx v3.3v
General: Fix typo in documentation. 414
1997-08-05 ltoutenc.dtx v1.9e
General: Corrected order of arguments in \UseTextSymbol example. . . . 214
1997-08-29 ltoutenc.dtx v1.9f
General: Added OT4 encoding, provided by Marcin Woliński. . . 213
1997-09-09 ltdefns.dtx v1.2z
\provide@command: Use \begingroup to avoid generating math ords if used in math mode. pr/2573 78
1997-09-15 ltpictur.dtx v1.1g
\@getcirc: Warn if lines become invisible pr/2524 573
\@picture@warn: Macro added pr/2524 573
\@sline: Warn if lines become invisible pr/2524 562
1997-10-06 ltcounts.dtx v1.1f
\@Roman: Change \@Roman to be fully expandable, so that the result is written properly to files. 263
\@slowromancap: Macro added. . . . 263
1997-10-08 ltlogos.dtx v1.1h
\LaTeX: Simplify macro (force loading of suitable math fonts once). . . . 192
1997-10-10 ltclass.dtx v1.0y
\endfilecontents: \@currenvir in banner 664
\reserved@c not \verbatim@out to save a csname 663
Check for text before or after \end environment. latex/2636 665
Use \@gobbletwo 663
1997-10-17 ltfntcmd.dtx v3.3w
\check@nocorr@: Check for vertical mode moved here, from \DeclareTextFontCommand (see PR/2646). 418
\DeclareTextFontCommand: Reinstalled \check@icr as check is now done in \check@nocorr@ (see PR/2646). 416
1997-10-20 ltfinal.dtx v1.1a
\@uclclist: Removed \aa and \AA from \@uclclist as these are macros. 808
1997-10-21 ltdefns.dtx v1.2z1
\renew@command: Use \begingroup/\endgroup rather than braces for grouping, to avoid generating empty math atom. . . . 76
1997-10-21 ltfssbas.dtx v3.0t
\define@newfont: Move \makeatletter to \nfss@catcodes. 281

\nfss@catcodes: Moved \makeatletter from \try@load@font@shape. 282
1997-11-09 ltoutput.dtx v1.2c
\@specialoutput: Remove incorrect code: only one \@emptycol is needed here 740
\@topnewpage: Documentation of vsize check enhanced 737
1997-11-13 ltfssdcl.dtx v3.0f
\DeclareSymbolFont: (DPC) Really update \group@list dont leave new version in \toks@. latex/2661 349
\stepcounter: (DPC) Remove as never used. (Re)defined in ltcounts 342
1997-11-19 ltfloat.dtx v1.1t
\@footnotetext: Missing percent, again 620
1997-11-19 ltoutput.dtx v1.2d
\@vtryfc: Reindent code, to be understandable(DPC). 757
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) 345
\select@group: (DPC) inline use of \stepcounter (faster, and saves a csname per math version as no reset list) 343
1997-11-23 ltoutenc.dtx v1.9g
General: Use \textperthousand, \textpertenthousand and \textfractionsolidus not \textpermill, \textpertenmill and \textfraction. /2673 240
1997-12-17 ltoutenc.dtx v1.9h
General: Added \textperthousand and \textpertenthousand . 231, 232
Added code for textcomp.sty. . . . 255
Added section. 255
Added textcomp.sty. 213
As in OT1, Added \leavevmode at start of \c, otherwise the output routine might be invoked within the macro. 232
Changed to decimal codes in \ooalign. 242
Changed to decimal codes. 238
Documentation changes and additions. 213
Example corrected, braces removed. 213
Removed default settings, see next section. 240
1997-12-19 ltoutenc.dtx v1.9i
General: Documentation corrections. 213
1997-12-20 fontdef.dtx v2.2s
General: Added documentation . . . 392
1997-12-31 ltoutenc.dtx v1.9k
General: Further correction 214
1998-01-12 ltoutenc.dtx v1.9k
General: Added \ProvidesPackage for textcomp.sty 213
Adding missing braces and \ushape. 242
1998-01-16 ltoutenc.dtx v1.9m
General: fixed decimal codes. latex/2734 238
1998-03-04 ltdefns.dtx v1.2z2
\@xargdef: Unnecessary \expandafter removed: pr/2758 . 74
1998-03-05 ltoutenc.dtx v1.9n
General: Added masc/fem ords as in pr/2579 227
1998-03-20 ltdefns.dtx v1.2z3
\@thirdofthree: Macro added 79
1998-03-20 ltoutenc.dtx v1.9o
General: Documentation added about order of decls 216
Documentation added for pr/2783 215
\UndeclareTextCommand: Macro added for pr/2783 224
1998-03-20 lttextcomp.dtx v1.9o
General: Added various \UndeclareTextCommand declarations for pr/2783 450
Load decls after defaults for speed. 450
1998-03-21 ltclass.dtx v1.0z
General: Added to documentation of filecontents 633
1998-03-21 ltclass.dtx v1.1a
\@providesfile: Allow &. Internal/2702 646
General: Correct to new onlypreamble command list 678
1998-03-25 ltfssbas.dtx v3.0u
\showhyphens: Suppress unnecessary error when used in preamble . . . 291
1998-04-11 fontdef.dtx v2.2t
General: Added \mathring accent (pr2785) 406
1998-04-15 fontdef.dtx v2.2u
General: Use new syntax for \DeclareMathDelimiter 399

1998-04-15 ltfssdcl.dtx v3.0h
`\@xxDeclareMathDelimiter`: Macro added (pr/2662) 359
1998-04-17 fontdef.dtx v2.2v
General: Reinsert symbol defs for `<` and `>` chars. 400
1998-04-18 fontdef.dtx v2.2w
General: Reinsert symbol def for `/` char. 400
1998-05-07 ltclass.dtx v1.1b
`\load@onefilewithoptions`: Modify help message for latex/2805 655
1998-05-18 lttab.dtx v1.1j
`\@endpbox`: Use `\setlength` to set `\hsize`, so that the changes in the calc package apply here. 550
`\tabular*`: Use `\setlength`, so that calc extensions apply. 539
1998-05-20 ltfinal.dtx v1.1b
General: Set up lccodes before loading hyphenation files: pr/2639 801
Set up uc/lccodes after loading hyphenation files: pr/2639 807
1998-05-28 lterror.dtx v1.2n
`\@notdefinable`: Added message re 'end...' pr/1555 167
1998-06-04 ltboxes.dtx v1.1c
`\@rule`: Support calc-expressions . . 524
1998-06-12 ltoutenc.dtx v1.9p
General: Corrected 130 and 131, see pr/2834 242
Renamed `\textmacron` pr/2840 . 243
1998-06-12 ltoutenc.dtx v1.9q
`\add@accent`: Explicitly set `\spacefactor` after `\accent` (pr/2877) 220
1998-06-12 lttextcomp.dtx v1.9p
General: Renamed `\textmacron` pr/2840 447
1998-06-18 lttab.dtx v1.1k
General: Small addition to documentation 527
1998-07-06 lttab.dtx v1.1l
General: Small correction to documentation 527
1998-08-17 ltboxes.dtx v1.1e
General: (RmS) Minor Documentation fixes. 512
1998-08-17 ltclass.dtx v1.1c
General: (RmS) Minor documentation fixes. 633
1998-08-17 ltdirchk.dtx v1.0w
General: (RmS) Documentation improvements. 1
1998-08-17 ltfntcmd.dtx v3.3x
General: (RmS) Minor documentation fixes. 414
1998-08-17 ltfssbas.dtx v3.0v
General: (RmS) Documentation fixes. 268
1998-08-17 ltfssdcl.dtx v3.0i
General: (RmS) Corrected minor glitches in changes entries. 340
1998-08-17 ltfssini.dtx v3.0i
General: (RmS) Minor documentation fixes. 367
1998-08-17 ltlogos.dtx v1.1i
General: (RmS) Minor documentation fixes. 192
1998-08-17 ltmath.dtx v1.1c
General: (RmS) Minor documentation fixes. 481
1998-08-17 ltmiscen.dtx v1.1g
General: (RmS) Minor documentation fixes. 460
1998-08-17 ltspace.dtx v1.2w
General: Documentation fixes. 173
1998-08-17 preload.dtx v2.1g
General: (RmS) Minor documentation fixes. 411
1998-09-19 ltoutenc.dtx v1.9r
`\a`: Added `\string` (pr/2878) 225
1998-11-13 lttab.dtx v1.1m
`\@array`: Check for hmode to see if something went wrong during parsing (pr/2884) 540
1999-01-05 fontdef.dtx v2.2x
General: Need special protection for character `>` in `\changes` entry. . . 390
1999-01-06 ltfssbas.dtx v3.0w
`\DeclareFontEncoding`: Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) 272
`\LastDeclaredEncoding`: Added `\LastDeclaredEncoding` to support cyrillic integration (pr/2988) 272
1999-01-06 ltoutenc.dtx v1.9r
`\@strip@args`: New impl for latex/2930 222
General: Minor documentation fix. . 242
1999-01-06 ltoutput.dtx v1.2e
`\@makecol`: Added negative vskip, as when processing outputbox below: suggested by Fred Bartlett pr/2892 746
1999-01-07 ltdefns.dtx v1.3a
`\@ifnextchar`: made long 95

\@newenvb: made long and brace optional arg. latex/2896 77
\@testopt: made long and brace optional arg. latex/2896 74
1999-01-07 ltdefns.dtx v1.3b
\@ifnextchar: extra \long. latex/2902 95
1999-01-07 ltoutenc.dtx v1.9r
General: Hackery to allow using fontenc several times 257
Hackery to temp support cyrillic uc/lc 255
1999-01-13 ltoutenc.dtx v1.9s
\@strip@args: Simplified solution for latex/2930 222
1999-01-18 ltdefns.dtx v1.3c
\@yargd@f: New implementation DPC /2942 75
1999-02-09 ltdefns.dtx v1.3d
\@yargd@f: catch bad argument forms by re-inserting #3 75
1999-02-12 lttextcomp.dtx v3.0j
\legacyoldstylenums: Use \rmdefault instead of cmm (pr/2954) 423
1999-02-24 ltoutenc.dtx v1.9t
General: Corrected hackery cyrillic uc/lc list 255
1999-03-01 ltdefns.dtx v1.3e
\@ifnextchar: remove extra \long. internal/2967 95
1999-04-15 ltpictur.dtx v1.1h
\@getlarrow: Replaced octal number, CAR 564
\@upvector: Replaced octal number, CAR 565
General: Replaced octal number, CAR 564, 565
Replaced octal numbers, CAR . . 552
1999-04-19 ltfloat.dtx v1.1u
\caption: Made caption an error outside a float: latex/2815 604
1999-04-27 ltboxes.dtx v1.1f
\@parboxto: (CAR) Changed \@empty to \relax as flag for natural width: pr/2975 520
1999-04-29 ltdefns.dtx v1.3f
\@yargd@f: Full expansion and conversion needed for digit in new version, see pr/3013 75
New macro added 75
1999-06-10 ltoutenc.dtx v1.9u
General: Ensure that we also forget old options (pr/2888) 257
1999-06-12 ltoutenc.dtx v1.9v
General: Extend \@uclclist only once 256
1999-10-09 ltmath.dtx v1.1e
\active@math@prime: Macro added, see PR 3104. 487
\prime@s: Introduce \active@math@prime. 488
1999-10-09 ltoutput.dtx 1.2f
\@activechar@info: Reset definition of active prime character (used in math mode) 749
1999-10-28 ltoutenc.dtx v1.9w
\add@accent: Give \accent@spacefactor a default definition (pr/3084) 220
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. 255
2000-01-07 ltmiscen.dtx v1.1h
\@verbatim: Disable hyphenation even if the font allows it. 475
2000-01-15 ltpictur.dtx v1.1i
\@upvector: Removed space at end-of-line, CAR 565
2000-01-30 ltfntcmd.dtx v3.3y
\DeclareTextFontCommand: Use \hmode@bgroup now (pr/3160) . . 416
2000-01-30 ltoutenc.dtx v1.9y
General: Use \hmode@bgroup where applicable (pr/3160) . . . 229–232, 237–240, 242
\add@accent: Use \hmode@bgroup where applicable (pr/3160) 219
\hmode@bgroup: Macro added 220
2000-01-30 ltoutenc.dtx v1.9z
\@use@text@encoding: Macro reimplemented (pr/3160) . . 222, 223
\add@accent: Macro reimplemented (pr/3160) 219
\hmode@start@before@group: Macro added (pr/3160) 223
2000-05-19 ltmiscen.dtx v1.1i
\enddocument: Reset \AtEndDocument for latex/3060 461
2000-05-26 ltpage.dtx v1.0j
\@markright: Reimplementation to fix expansion error (pr/3203). 631
\leftmark: Use \@empty instead of brace group (pr/3203). 631

\markright: Reimplementation to fix expansion error (pr/3203). 630
\rightmark: Use \@empty instead of brace group (pr/3203). 631
2000-06-02 ltpage.dtx v1.0k
\@markright: Small adjustment to give slightly less expansion, CAR 631
\markright: Small adjustment to give slightly less expansion, CAR . . . 630
Tidied 1.0j reimplementation, CAR 630
2000-07-11 ltmiscen.dtx v1.1j
\@enddocument@kernel@warnings: Fix typo in warning 463
2000-07-12 ltoutput.dtx 1.2g
General: Ensure that rule is in \normalcolor 789
2000-07-12 ltoutput.dtx 1.2i
\@makecol: Removed negative vskip, as it gives unacceptable results when the depth is large: pr/3189 746
2000-07-19 ltoutput.dtx v1.2h
\@writesetup: Reset and restore \@if@newlist for internal/3231 . 750
2000-08-23 ltfinal.dtx v1.1c
General: Fix typo in warning 803
2000-08-30 ltoutenc.dtx v1.91
\@use@text@encoding: Rearranged but no change to final code, CAR (pr/3160) 222
\add@accent: Rearranged but no change to final code, CAR (pr/3160) 219
2000-09-01 ltfinal.dtx v1.1d
\errhelp: Set error help empty at very end (pr/449 done correctly). 811
2000-09-24 ltfloat.dtx v1.2b
\end@dblfloat: FMi: use output routine to defer float 609
2000-09-24 ltoutput.dtx v1.2b
\@doclearpage: FMi: ensure \doclearpage is called again until all floats are output. 744
2000-09-24 ltoutput.dtx v1.2n
\@addtocurcol: FMi: test for wide float was in wrong place 762
2001-01-07 ltoutput.dtx v1.2j
\@writesetup: And do it in the right macro (pr/3286) 750
2001-02-16 ltxref.dtx v1.1k
\@newl@bel: Added an extra grouplevel (PR3250), jlb 457
2001-05-25 ltclass.dtx v1.1d
\@providesfile: Explicitly set catcode of \endlinechar to 10 (pr/3334) 646
2001-05-25 ltdirchk.dtx v1.0x
General: Explicitly set catcode of \endlinechar to 10 (pr/3334) . . . 4
2001-05-28 ltoutenc.dtx v1.93
General: Added composites for compatibility with T1, pr/3295 . 231
Changed the effect of \.\i, pr/3295 234
2001-06-02 fontdef.dtx v2.2y
General: Provide default cfg files (pr/3264) 409
2001-06-04 fontdef.dtx v2.2z
General: Guard against math active equal and pipe sign in \models (pr/3333) 405
Guard against math active equal sign in \Relbar (pr/3333) 405
2001-06-04 ltclass.dtx v1.1e
\@providesfile: But only if it is a char (pr/3334) 646
2001-06-04 ltdirchk.dtx v1.0y
General: But only if it is a char (pr/3334) 4
2001-06-04 ltpictur.dtx v1.1j
\@sline: Don't warn for exactly zero pr/3318 562
2001-06-04 ltvers.dtx v1.0i
General: Check for old format disabled 34
2001-06-05 ltoutenc.dtx v1.94
General: Text composite Commands need kludges for ',' – see tlb1903.lvt 231
2001-08-26 ltclass.dtx v1.1f
\@providesfile: Readded setting of space char (pr/3353) 646
2002-02-24 ltplain.dtx v1.1x
\loggingall: Macro added 31
\loggingoutput: Macro added 31
\showoutput: Use newly added \loggingoutput 31
\tracingall: Use newly added \loggingoutput 31
2002-06-16 ltoutenc.dtx v1.95
General: Added \textbardbl (pr/3400) 237
Added default for \textbardbl (pr/3400) 226
2002-06-17 ltoutenc.dtx v1.95
General: Corrected \c for T1 (pr/3442) 232
Definition of \textexclamdown changed (pr/3368) 230

Definition of \textquestiondown changed (pr/3368) 230
2002-06-18 ltoutenc.dtx v1.95
General: Changed def for \textregistered to avoid small caps (pr/3420) 227
2002-10-01 ltfloat.dtx v1.1v
\thempfootnote: Use braces around \itshape to keep font change local (pr/3460). 618
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) 272
2002-10-28 ltlists.dtx v1.0s
\endtrivlist: Check for math mode (pr/3437) 506
2002-10-28 ltoutenc.dtx v1.96
General: coding change, to follow bug fix by DEK in plain.tex (pr/3469) 230, 239
2002-12-13 ltbibl.dtx v1.1n
\@citex: Added \leavevmode in case citation is at start of paragraph (pr/3486) 626
2003-01-01 ltfntcmd.dtx v3.3z
General: Code checked and documentation extended by Chris 416
2003-05-18 ltbibl.dtx v1.1o
\nocite: Check if we are after \document 628
2003-08-27 ltpictur.dtx v1.1k
\@bezier: added missing displacement pr/3566 581
\@sline: check for \@linechar being empty pr/3570 562
2003-10-13 ltfinal.dtx v1.1e
General: Added extra \lccode for \- and \textcompwordmark 802
2003-12-16 ltoutput.dtx v1.2k
\@makecol: Ensure that \@elt has a defined state (pr/3586) 746
2003-12-30 ltpictur.dtx v1.1j
\@getcirc: issue warning if circle size can't be met pr/3473 573
2004-01-03 ltoutenc.dtx v1.99b
General: Added \textogonekcentered (pr/3532) 232
Added composites for \k (pr/3532) 236
Use \ooalign for \k (pr/3532) . . 232
2004-01-04 ltbibl.dtx v1.1p
\nocite: Changed error message . . 628
2004-01-04 ltoutenc.dtx v1.99c
General: More adjustments for ogonek (pr/3532) 232
2004-01-23 ltdefns.dtx v1.1g
\@newenva: Use kernel version of \@ifnextchar (pr/3501) 77
\@testopt: Use kernel version of \@ifnextchar (pr/3501) 74
\@xargdef: Use kernel version of \@ifnextchar (pr/3501) 74
\@xdblarg: Use kernel version of \@ifnextchar (pr/3501) 96
2004-01-23 ltdefns.dtx v1.3g
\kernel@ifnextchar: Added macro (pr/3501) 95
2004-01-28 ltclass.dtx v1.1g
\@providesfile: Use kernel version of \@ifnextchar (pr/3501) 646
2004-01-28 ltvers.dtx v1.0k
General: Check for old format made 5 years (pr/3601) 34
2004-02-02 fontdef.dtx v2.3
General: Many things from here on made robust 404
2004-02-02 ltoutenc.dtx v1.99
General: Added \textbigcircle . . 237
2004-02-04 fontdef.dtx v2.3a
General: Added bigtriangle synonyms for stmaryrd 402
2004-02-04 ltspace.dtx v1.3
\nobreakdashes: (Macro added . . . 188
2004-02-06 ltoutenc.dtx v1.99d
\@inmathwarn: New command added to fix severe bug: pr/3563 217
2004-02-07 ltoutput.dtx v1.2l
\@doclearpage: Empty kludgeins box if necessary, pr/3528 744
2004-02-13 ltoutenc.dtx v1.99e
General: Documentation fixes: typos 213
2004-02-15 ltbibl.dtx v1.1q
\@cite@ofmt: Added hook with default value \hbox 628
\@citex: Changed to use a hook with default value \hbox 627
2004-02-15 ltspace.dtx v1.3a
\nobreakdashes: (Added spacefactor setting 188
2004-10-20 ltoutput.dtx v1.2m
\@makecol: Removed dead code . . . 746
2005-07-27 ltfssdcl.dtx v3.0j
\DeclareMathAlphabet: (MH) Make document commands robust . . . 351

\DeclareSymbolFontAlphabet: (MH) Make document commands robust 364
\new@mathalphabet: (MH) Make document commands robust . . . 352
\non@alpherr: (MH) Change because command is now properly robust 344
\SetMathAlphabet: (MH) Make document commands robust . . . 353
2005-09-27 ltoutenc.dtx v1.99g
General: Replace \sh@ft by \ltx@sh@ft 229, 232, 238
2005-09-27 ltplain.dtx v1.1y
\ltx@sh@ft: New macro 30
\sh@ft: Macro no longer used but left for compatibility 30
2005-11-08 ltoutenc.dtx v1.99h
General: Added \ij and \IJ from babel. (pr/3771) 226, 230, 233
2005-11-10 ltmath.dtx v1.1g
\[: (MH) Fixed potential problem in \[(pr/3399). 488
General: (MH) Minor documentation fixes. 481
2006-05-18 ltboxes.dtx v1.1g
\@parboxto: Ensure \@parboxto holds the value of \@tempdimb not the register itself (pr/3867) 520
2006-09-13 ltoutput.dtx v1.1m
General: Ensure that rule is in \normalcolor 790
2007-08-05 ltclass.dtx v1.1h
\@fileswithoptions: Prevent loss of brackets PR/3965 653
2007-08-06 ltcntrl.dtx v1.0h
\@fornoop: Really make defs long . . 160
2007-08-31 ltfssdcl.dtx v3.0l
\SetSymbolFont@: Font warning changed to info for encoding change (pr/3975) 350
2009-09-24 ltvers.dtx v1.0l
General: Stop checking for old format 34
2009-10-20 ltfssdcl.dtx v3.0m
\in@: More robust thanks to Heiko. 340
2009-10-28 lttextcomp.dtx v1.99k
General: Added Latin Modern and TeX Gyre subsets 452
2009-11-04 lttextcomp.dtx v1.99l
General: Added more Latin Modern and TeX Gyre subsets 452
2009-12-14 ltfntcmd.dtx v3.4a
\ifmaybe@ic: Macro added 419
\maybe@ic@: Use switch \ifmaybe@ic instead of \if@tempswa 419
\t@st@ic: Use switch \ifmaybe@ic instead of \if@tempswa 420
2010-08-17 ltmiscen.dtx v1.1k
\enddocument: Use braces around \input arg (pr/4124) 462
2010-08-17 ltmiscen.dtx v1.1l
\enddocument: Change of plan: use \@@input instead (pr/4124) 462
2011-05-08 ltfssdcl.dtx v3.0n
\in@: Simplified thanks to Bruno. . . 340
2011-08-19 ltclass.dtx v1.1i
\@ifclasswith: Re-jig definition after more stringent \in@ test. 643
2011-09-03 ltfssdcl.dtx v3.0o
\new@mathversion: (Will) Remove \global before \newcount (unnecessary and caused etex bug). 348
2012-01-20 ltplain.dtx v2.0b
\loggingall: etex tracing if available 31
2013-07-07 ltclass.dtx v1.1i
General: Correctly describe how the date in \@ifpackagelater is used 636
2014-04-18 ltoutput.dtx v1.1o
General: Handle infinite glue from \enlargethispage (pr/4023) . . . 790
2014-04-24 ltoutput.dtx v1.2n
\fl@tracemessage: Renamed internal trace commands; provide as package 780
2014-04-27 ltfloat.dtx v1.2b
\end@dblfloat: Inline the code to allow some coexistence with packages that hook into \end@float and do not know about the algorithm change 609
2014-06-10 ltfloat.dtx v1.2b
\end@dblfloat: missing \fi added . 609
2014-12-30 ltfinal.dtx v2.0a
\newmarks: macro added 797
\newXeTeXintercharclass: macro added 797
2014-12-30 ltfloat.dtx v1.2a
\@textsubscript: Command added (latexrelease) 619
\textsubscript: Command added (latexrelease) 619
2014-12-30 ltfssbas.dtx v3.0y
\mathgroup: move allocation to ltplain. 268
2014-12-30 ltoutput.dtx v1.2m
General: Command updated (latexrelease) 789

2014-12-30 ltplain.dtx v2.0a
\e@alloc: macro added 19
\e@alloc@chardef: macro added . . . 18
\e@alloc@top: macro added 18
\e@ch@ck: macro added 19
\extrafloats: macro added 20
\newlanguage: New engine-specific allocation scheme (latexrelease) . . 17
2014-12-30 ltspace.dtx v1.3b
\@: \@ discards spaces when moving (pr3039)(latexrelease) 189
2015-01-03 ltdefns.dtx v1.4a
\typein: use modified definition in luatex 72
2015-01-03 ltdirchk.dtx v1.1
General: Enable extra primitives when LuaTEX is used 3
2015-01-03 ltfinal.dtx v2.0a
General: Skip resetting codes with Unicode engines 807
Unicode data loading added 799
2015-01-07 ltvers.dtx v1.0n
\IncludeInRelease: macro added . . 35
2015-01-08 ltboxes.dtx v1.1h
\framebox: Make Robust (latexrelease) 518
\makebox: Make Robust (latexrelease) 513
\parbox: Make Robust (latexrelease) 520
\raisebox: Make Robust (latexrelease) 525
\rule: Make Robust (latexrelease) . 524
\savebox: Make Robust (latexrelease) 516
2015-01-08 ltdefns.dtx v1.4a
\MakeRobust: Added macro 82
2015-01-08 ltlength.dtx v1.1c
\setlength: to ensure first length argument is terminated. (latexrelease) 266
2015-01-08 ltmath.dtx v1.1h
\): Make Robust (latexrelease) 488
\]: Make Robust (latexrelease) 488
2015-01-09 ltfssini.dtx v3.1a
\em: Allow \emph to produce small caps (latexrelease) 384
\eminnershape: macro added (latexrelease) 384
2015-01-09 ltspace.dtx v1.1h
\addpenalty: Donald Arseneau's fix from PR/377703 (latexrelease) . . 185
2015-01-10 ltcounts.dtx v1.1h
\@fnsymbol: Unse \TextOrMath (latexrelease) 263
\@stpelt: Reset all within counters in one go (latexrelease) 260
2015-01-11 ltcounts.dtx v1.1h
\TextOrMath: Add command to solve robustness issues (pr/3752) (latexrelease) 264
2015-01-11 ltfloat.dtx v1.2b
\@dblfloatplacement: float order in 2-column (latexrelease) 611
\@xfloat: Check for valid option (latexrelease) 605
\end@dblfloat: float order in 2-column (latexrelease) 609
2015-01-11 ltfssbas.dtx v3.0y
\@DeclareMathSizes: Allow arbitrary units (latexrelease) 275
2015-01-11 ltspace.dtx v1.3d
\@Esphack: Allow hyphenation (Donald Arseneau pr/3498) (latexrelease) 182
\@esphack: Allow hyphenation (Donald Arseneau pr/3498) (latexrelease) 181
2015-01-14 ltoutput.dtx v1.2n
\@addtocurcol: float order in 2-column (latexrelease) 761
\@addtodblcol: float order in 2-column (latexrelease) 772
\@addtonextcol: float order in 2-column (latexrelease) 768
\@doclearpage: Empty kludgeins box if necessary, pr/3528 743
float order in 2-column (latexrelease) 743
\@startdblcolumn: float order in 2-column (latexrelease) 755
\@xtryfc: float order in 2-column (latexrelease) 757
\@ztryfc: float order in 2-column (latexrelease) 758
2015-01-14 ltspace.dtx v1.3e
\addpenalty: Avoid adding redundant skips (DPC) 185
2015-01-17 ltvers.dtx v1.0m
\IncludeInRelease: modified with \@currname 35
2015-01-19 ltvers.dtx v1.0o
\IncludeInRelease: Optional argument 35
2015-01-20 ltoutput.dtx v1.2m
\fl@tracemessage: Reset \IncludeInRelease flags 781
2015-01-22 ltvers.dtx v1.0p
General: Preserve any \everyjob material inserted by a loader (.ini file) 35

2015-01-23 ltfinal.dtx v2.0b
\newmarks: use reserved count 256 . 797
\newXeTeXintercharclass: use reserved count 257 797
2015-01-23 ltplain.dtx v2.0c
\extrafloats: reserve counts 256–265 20
2015-01-24 ltfinal.dtx v2.0c
General: Skip T1-code entirely with Unicode engines 799
2015-02-03 ltfinal.dtx v2.0d
General: Set \lccode for - with Unicode engines 800
2015-02-16 ltoutenc.dtx v1.99m
General: Added \textcommabelow latex/4414 228
2015-02-16 ltoutenc.dtx v1.99n
General: Added \textcommaabove . . 229
Added composites for ç 236
Added composites for \c 231
2015-02-16 lttextcomp.dtx v1.99m
General: Added lmtt (Heiko Oberdiek) latex/4415 452
2015-02-19 ltvers.dtx v1.0q
\IncludeInRelease: Swap argument order 35
2015-02-20 ltplain.dtx v2.0d
\loggingall: Spell commands correctly :-) 31
2015-02-21 ltdefns.dtx v1.4b
General: Removed autoload support . 70
2015-02-21 lterror.dtx v1.2o
General: Removed autoload support 161
2015-02-21 ltfiles.dtx v1.1m
General: Removed autoload support 193
2015-02-21 ltfssbas.dtx v3.0z
General: Removed autoload code . . 268
2015-02-21 ltfsscmp.dtx v3.0d
General: Removed autoload code . . 335
2015-02-21 ltfssdcl.dtx v3.0p
General: Removed autoload code . . 340
2015-02-21 ltfsstrc.dtx v3.0k
General: Removed autoload code . . 312
2015-02-21 ltoutenc.dtx v1.99m
General: Removed autoload code . . 213
2015-02-21 ltoutput.dtx v1.2n
General: Removed autoload code . . 722
\f@depth: macro added(latexrelease) 742
2015-02-21 ltpictur.dtx v1.1k
General: Removed autoload code . . 552
2015-02-21 ltplain.dtx v2.0e
General: Removed autoload code . . . 14
2015-02-21 lttab.dtx v1.1n
General: Removed autoload code . . 527
2015-02-21 ltvers.dtx v1.0r
General: Removed autoload code . . . 34
2015-02-21 ltvers.dtx v1.0w
\IncludeInRelease: set \@currname empty here (in case \IncludeInRelease input early) . 35
2015-02-22 ltfsscmp.dtx v3.0e
General: Moved all code into latexrelease - obsolete commands are no longer automatically part of the kernel 335
2015-03-02 ltplain.dtx v2.0f
\e@mathgroup@top: macro added . . . 19
\newlanguage: allow 255 math groups in Unicode engines 17
2015-03-10 ltplain.dtx v2.0g
\hideoutput: macro added 32
\loggingall: Reorganise to be less noisy 31
\tracingnone: macro added 32
2015-03-12 ltoutput.dtx v1.2m
General: initialise \@dbldeferlist again 733
2015-03-18 ltfssdcl.dtx v3.0q
\DeclareSymbolFont: Restrict Symbol fonts to 0-15 349
\document@select@group: Introduce \e@mathgroup@top 345
\select@group: Introduce \e@mathgroup@top 343
2015-03-26 ltfinal.dtx v2.0d
General: Use renamed unicode-letters.def 799
2015-04-07 ltfssbas.dtx v3.1a
\wrong@fontshape: Try loading fd file if family has changed 285
2015-04-28 ltfinal.dtx v2.0f
\newXeTeXintercharclass: define \xe@alloc@intercharclass for compatibility with older xelatex initilisation 797
2015-05-10 ltlists.dtx v1.0t
\@doendpe: Explicitly reset \clubpenalty before clearing \everypar; see also pr/0462 and pr/4065 506
2015-06-19 ltfinal.dtx v2.0g
\e@alloc@intercharclass@top: Use −1 for first range to get contiguous allocation 798
\newmarks: Use −1 for first range to get contiguous allocation 797

2015-06-19 ltplain.dtx v2.0h
General: delete spurious old definition of \newtoks 23
\e@alloc: extra braces in case arguments not single token 19
\newlanguage: Use −1 for first range to get contiguous allocation 17
2015-06-23 ltfinal.dtx v2.0h
General: set \patch@level in ltvers rather than in ltfinal/ltpatch . . . 809
2015-06-23 ltvers.dtx v1.0t
General: set \patch@level in ltvers rather than in ltfinal/ltpatch 34
2015-08-06 ltplain.dtx v2.0i
\extrafloats: Add \string in case argument is not an unexpandable primitive 20
2015-08-23 ltdirchk.dtx v1.2
General: Do not use luatex prefix 3
2015-08-23 ltvers.dtx v1.0v
General: Allow negative patchlevel for pre-release 35
2015-08-30 ltplain.dtx v2.1a
\newinsert: new \newinsert implementation 22
2015-09-205 ltoutput.dtx v1.3a
General: extended \@freelist 732
2015-09-24 ltluatex.dtx v1.0a
call_callback: Function added 60
callback.register: Function modified 57
callback_descriptions: Function added 63
\catcodetable@atletter: Macro added 45
\catcodetable@initex: Macro added 45
\catcodetable@latex: Macro added . 45
\catcodetable@string: Macro added 45
add_to_callback: Function added . . 61
remove_from_callback: Function added 62
new_attribute: Function added 52
disable_callback: Function added . 63
in_callback: Function added 63
\newattribute: Macro added 44
\newcatcodetable: Macro added . . . 44
\newluabytecode: Macro added 47
\newluachunkname: Macro added . . . 47
\newluafunction: Macro added 46
\newwhatsit: Macro added 47
module_error: Function added 51
module_info: Function added 51
module_warning: Function added . . . 51
modules: Function modified 49
create_callback: Function added . . 60
provides_module: Function added . . 50
luatexbase: Table added 49
2015-10-02 ltdirchk.dtx v1.2a
General: Allow backing out of unprefixed names 3
2015-10-02 ltluatex.dtx v1.0b
General: Fix backing out of TEX code 48
2015-10-02 ltluatex.dtx v1.0c
General: Allow backing out of Lua code 48
2015-10-02 ltluatex.dtx v1.0e
uninstall: Function added 63
2015-10-03 ltluatex.dtx v1.0f
provides_module: use luatexbase_ log 50
2015-10-27 ltplain.dtx v2.1b
\extrafloats: Use global assignment when switching to extended range 20
2015-11-07 ltspace.dtx v1.3f
\@esphack: Only space if there is no space at the end of the hlist latex/4443 181
2015-11-14 ltluatex.dtx v1.0g
General: Track LuaTEX changes for (new)token.create 51
2015-11-18 ltplain.dtx v2.2a
\newlanguage: Extended stream allocation in luatex (0.85) 17
2015-11-19 ltplain.dtx v2.2b
\newlanguage: Only extend allocation of write streams (see luatex list) . 17
2015-11-27 ltluatex.dtx v1.0h
callback_descriptions: Match test in in-callback latex/4445 63
in_callback: Guard against undefined list latex/4445 63
2015-11-29 ltluatex.dtx v1.0i
General: Declare this as local before used in the module error definitions (PHG) 49
call_callback: Check name is not nil in error message (PHG) 60
create_callback: Check name is not nil in error message (PHG) 60
2015-12-02 ltluatex.dtx v1.0j
General: Adjust hashtokens to store the result of tex.hashtokens()), not the function (PHG) 51
Assorted typos fixed (PHG) 42
Declaration/use of first_ head fixed (PHG) 50
Remove nonlocal iteration variables (PHG) 42

Remove unreachable code after calls to error() (PHG) 42
2015-12-02 ltluatex.dtx v1.0k
General: resolve name and i.description (PHG) 58
`call_callback`: Give more specific error messages (PHG) 60
`add_to_callback`: Give more specific error messages (PHG) 61
`remove_from_callback`: adjust initialisation of cb local (PHG) . . 62
Give more specific error messages (PHG) 62
`create_callback`: Give more specific error messages (PHG) 60
2015-12-10 ltfinal.dtx v2.0i
General: Use new common Unicode data loaders 799
2015-12-18 ltluatex.dtx v1.0l
General: Load Unicode data from source 45
2016-01-04 ltfinal.dtx v2.0j
General: Do not set up inter character classes for XeTeX 799
`\e@alloc@intercharclass@top`: Start allocation at one not three 797
2016-01-05 ltfinal.dtx v2.0k
`\e@alloc@intercharclass@top`: Remove duplicated code 797
2016-01-05 ltfinal.dtx v2.0l
General: Correct latexrelease guards 799
Ensure old definitions for inter-character class toks are available using latexrelease 799
Missing brace 799
2016-01-05 ltfinal.dtx v2.0m
General: Undefine XeTeX classes when using patching an older kernel . . 799
2016-01-05 ltfinal.dtx v2.0p
General: Only apply XeTeX change if XeTeX is in use 799
2016-02-11 ltluatex.dtx v1.0m
General:
pdf_ stream_ filter_ callback removed 57
process_ rule, [hv]pack_ quality append_ to_ vlist_ filter added . 56
read_ cidmap_ file added 56
show_ warning_ message added . . 56
token_ filter removed 56
2016-02-18 ltfssdcl.dtx v3.0r
`\@DeclareMathDelimiter`: Check for delimiter not `\delimiter` 360
`\DeclareMathAccent`: Check for mathaccent not `\mathaccemt` . . . 355
`\DeclareMathRadical`: Check for radical not `\radical` 362
`\DeclareMathSymbol`: Check for mathchar not `\mathchar` 357
2016-03-13 ltluatex.dtx v1.0n
General: contribute_ filter added . . . 56
insert_ local_ par added 56
2016-03-29 ltpictur.dtx v1.1l
`\@oval`: add setting of line tests 574, 575
initialise tests 574
`\@ovhorz`: use glue not leaders if horizontal line not required 577
`\@ovvert`: use glue not leaders if vertical line not required 576
`\if@ovhline`: macro added (latex/4452) 574
`\if@ovvline`: macro added (latex/4452) 574
2016-04-22 ltfinal.dtx v2.0q
`\e@alloc@intercharclass@top`: XeTeX 0.99996 has 4096 char classes not 256 798
2016-06-19 ltoutenc.dtx v1.99m
General: OT1 definition (was duplicate T1 definition) 231
2016-06-20 ltclass.dtx v1.1j
General: don't declare as `\@onlypreamble` 642
2016-07-29 ltplain.dtx v2.2c
`\extrafloats`: use `\global \chardef` 20
`\newinsert`: fix for tlb-newinsert-001 22
2016-10-02 ltclass.dtx v1.2a
`\@ifclasswith`: Ignore spaces while checking for option clash 643
`\ExecuteOptions`: Ignore spaces in argument 650
2016-10-15 ltdirchk.dtx v1.2b
General: Require eTEX 4
2016-10-15 lterror.dtx v1.2p
General: Require eTEX 161
2016-10-15 ltfinal.dtx v2.0r
General: Require eTEX 797
2016-10-15 ltfinal.dtx v2.0s
General: Tidy up status of char 127 797
2016-10-15 ltfssini.dtx v3.1b
General: Require eTEX 367
2016-10-15 ltplain.dtx v2.2d
General: Require eTEX 14
2016-10-16 ltplain.dtx v2.3a
`\newlanguage`: Allow languages up to 16383 in luatex 17

2016-10-19 ltcounts.dtx v1.1j
\TextOrMath: Test directly for \protected 264
2016-11-06 ltplain.dtx v2.3b
General: Drop \outer entirely 14
2016-11-09 ltclass.dtx v2.1b
\@fileswithoptions: Improve \ifx tests PR/4497 653
2016-11-17 ltluatex.dtx v1.0p
General: call_ edit added 56
2016-12-03 fontdef.dtx v3.0a
General: (DPC) Default to TU encoding for Unicode TeX engines 392
\shapedefault: (DPC) Default to TU encoding for Unicode TeX engines 396
2016-12-04 ltoutenc.dtx v2.0a
General: Added TU encoding 244
2017-01-01 ltoutput.dtx v1.3b
General: make fpmin negative so ignored even if float height is negative 789
2017-01-10 ltfssbas.dtx v3.2a
\showhyphens: Add version of \showhyphens that works with XeTEX. 291
2017-01-23 ltoutenc.dtx v2.0b
General: Added TU specific commands in ASCII range pr/4500 244
2017-01-24 ltoutenc.dtx v2.0c
General: Declare TU composites for i and j 244
Make \textasteriskcentered U+2217 not U+204E 244
TeX ligature syntax for xetex and luatex reversed 244
2017-01-24 ltoutenc.dtx v2.0d
General: Declare macron composites for YyGg 244
2017-02-12 ltoutenc.dtx v2.0e
General: Declare fallback code for \textasteriskcentered 244
2017-02-18 ltluatex.dtx v1.1c
new_attribute: Parameterise count used in tracking 52
new_bytecode: Parameterise count used in tracking 53
new_chunkname: Parameterise count used in tracking 54
new_whatsit: Parameterise count used in tracking 53
2017-02-19 ltoutenc.dtx v2.0f
General: add \@empty to guard against 3rd argument being empty 230
declare composites with empty base for hat and tilde, use same slots for \textasciicircum ans \textasciitilde 244
declare straight quotes using new \remove@tlig command 244
2017-02-22 ltoutenc.dtx v2.0g
General: Fix typo introduced at 2.0f 244
2017-02-24 ltoutenc.dtx v2.0h
General: introduce \DeclareUnicodeAccent 244
\DeclareTextCompositeCommand: add check whether the accent command is defined for this encoding 220
2017-03-08 ltclass.dtx v1.2c
General: add \@parse@version@dash to support yyyy-mm-dd as well as yyyy/mm/dd 642
2017-03-09 ltfinal.dtx v2.0t
\l@nohyphenation: ensure \l@nohyphenation is defined. . . 802
2017-03-09 ltmiscen.dtx v1.1m
\@verbatim: Use \language not \hyphenchar 475
\verb: Use \language to stop hyphenation 479
2017-03-10 ltfiles.dtx v1.1n
\document: Save language default . . 196
2017-03-10 ltoutput.dtx v1.3c
\@writesetup: Reset \language . . . 750
2017-03-13 ltdefns.dtx v1.5a
\-: Define \- in terms of \hyphenchar 97
2017-03-27 ltdefns.dtx v1.5b
\@dischyph: Define \@dischyph after \- 97
2017-03-28 ltluatex.dtx v1.1e
General: glyph_ stream_ provider added 57
2017-03-29 ltboxes.dtx v1.3a
\@arrayparboxrestore: Reset \lineskiplimit 521
2017-04-05 ltoutenc.dtx v2.0i
\DeclareTextCompositeCommand: Declare accent command if not already declared when declaring a composite. 220
2017-04-10 ltplain.dtx v2.3c
\newlanguage: Correction to code to skip write18 in luatex 17
2017-04-11 ltoutput.dtx v2.4a
\newpage: account for the depth of the last row of the page 736
2017-12-17 ltoutput.dtx v1.4b
\@addtonextcol: fix doc guards . . . 768

2018-01-06 ltdefns.dtx 1.5c
\\@ifundefined: Avoid defining undefined commands to \\relax . . 93
2018-02-18 ltclass.dtx v1.2d
\\@ifl@ter: Added 0 up front to make bad data come out as 0. 642
General: Introduce rollback concept 670
2018-03-08 ltcounts.dtx v1.1k
\\@ifbothcounters: Interface added 261
\\@removefromreset: Interface added 260
\\counterwithin: Interface added . . 261
\\counterwithout: Interface added . 261
2018-03-24 ltclass.dtx v1.2e
\\pkgcls@use@this@release: Use full file name for old release 676
2018-03-25 ltfinal.dtx v2.1a
General: default to UTF-8 803
\\UseRawInputEncoding: Macro added 804
2018-03-27 ltclass.dtx v1.2f
\\endfilecontents: Use full file name for old release 664
2018-04-06 ltfinal.dtx v2.1b
\\UseRawInputEncoding: Undo changes to \\DeclareFontEncoding@ and definition of \\DeclareUnicodeCharacter 804
2018-04-07 ltfinal.dtx v2.1c
\\UseRawInputEncoding: Undefine \\inputencodingname 804
2018-04-08 ltclass.dtx v1.2g
\\@ifl@ter: Strip leading spaces from dates. 642
2018-04-08 ltclass.dtx v1.2h
\\@onefilewithoptions: Pass expanded date 672
2018-04-08 ltfinal.dtx v2.1d
General: Delay full UTF-8 handling to \\everyjob 805
2018-04-11 ltcounts.dtx v1.1l
\\counterwithin: Correct default (issue/38) 261
2018-05-02 ltluatex.dtx v1.1g
General: find_ sfd_ file removed . . . 55
finish_ synctex_ callback added . . 56
glyph_ not_ found added 57
read_ sfd_ file removed 55
2018-05-08 ltclass.dtx v1.2i
\\pkgcls@parse@date@arg: Make suspicious rollback a warning not error: github issue 43 673
2018-05-11 ltfinal.dtx v2.18
General: Make invalid UTF-8 also safe, for legacy filesystem encodings . . 805
2018-05-29 ltclass.dtx v1.2j
\\endfilecontents: use \\csname not \\@undefined 665
2018-08-11 ltoutenc.dtx v2.0j
General: Provide \\guillemetleft and \\guillemetright 233, 239, 247, 248
2018-08-18 ltluatex.dtx v1.1h
General: append_ to_ vlist_ filter is exclusive 56
2018-08-24 ltfinal.dtx v2.1f
\\document@default@language: Add to latexrelease (github/68) 802
2018-09-02 ltsect.dtx v1.1b
\\@dottedtocline: Prevent protrusion (https://tex.stackexchange.com/q/172785/10109) 599
2018-09-24 fontdef.dtx v3.0b
General: Start LR-mode if necessary (git/49) 408
2018-09-24 ltmath.dtx v1.2b
\\smash: Start LR-mode if necessary (git/49) 484
\\vphantom: Start LR-mode if necessary (git/49) 484
2018-09-24 ltspace.dtx v1.3h
\\enspace: Start LR-mode if necessary (git/49) 190
\\leavevmode@ifvmode: Macro added (git/49) 190
2018-09-26 ltdefns.dtx v1.5e
\\renew@command: Always explicitly generate a space after the csname and not rely on \\noexpand to save tokens (git/41) 76
2018-09-26 ltmiscen.dtx v1.1n
\\@writefile: Sometimes mask the endline char when writing to files (github/73) 466
\\add@percent@to@temptokena: Sometimes mask the endline char when writing to files (github/73) 465
\\protected@file@percent: Sometimes mask the endline char when writing to files (github/73) 465
2018-09-26 ltsect.dtx v1.1c
\\addcontentsline: Sometimes mask the endline char when writing to files (github/73) 597
2018-10-10 ltspace.dtx v1.3i
\\@esphack: Don't introduce breakpoints if @nobreak is true and after sections 181

2018-10-11 ltmiscen.dtx v1.1o
\@@sverb: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 478
\@setupverbvisiblespace: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 477
\@verbvisiblespacebox: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 477
\asciispace: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 476
verbatim*: Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 477
\verbvisiblespace: Provide \verbvisiblespace such that it is usable in normal text (github/70) 476
Provide visible space in \verb* also for XeTeX and LuaTeX (github/69) 476
2018-10-21 ltluatex.dtx v1.1i
new_luafunction: Function added . . 54
2018-11-09 ltbibl.dtx LaTeX2e
\bibliography: Zap spaces in the argument as BibTeX doesn't support them (github/88) 627
2018-11-18 ltoutenc.dtx v2.0k
General: Provide \Hwithstroke and \hwithstroke 249
2018-11-19 ltoutenc.dtx v2.0k
General: Added \Hwithstroke and \hwithstroke 232
2018-11-28 ltoutput.dtx v1.4d
\@combinedblfloats: Unbox \@outputbox to preserve boxing level (github/94) 755
2018-12-30 lttab.dtx v1.1p
\@tabclassz: Add extra \hskip to guard against an \unskip at the start of a c-column cell (gh/102) 547
2019-02-07 ltfilehook.dtx v1.1o
\unqu@tefilef@und: Expand \@filef@und before executing second argument (github/109) . . 689
2019-02-07 ltfiles.dtx v1.1o
\@swaptwoargs: Helper macro added 208
2019-02-07 ltfssbas.dtx v3.2b
\define@newfont: Changed wording of warning (github/107) 281
2019-06-18 ltluatex.dtx v1.1j
General: finish_ synctex_ callback renamed finish_ synctex 56
font_ descriptor_ objnum_- provider added 57
make_ extensible added 57
new_ graf added 56
page_ objnum_ provider added . . 56
process_ pdf_ image_ content added 56
wrapup_ run added 56
2019-07-01 ltclass.dtx v1.3a
\endfilecontents: Support UTF8 and spaces in filecontents environment file name 662
2019-07-01 ltfiles.dtx v1.2a
\IfFileExists: Support UTF-8 . . . 206
\includeonly: Support UTF-8 201
\set@curr@file: Support UTF-8 . . 205
2019-07-09 ltfssbas.dtx v3.2c
\DeclareErrorFont: Don't set any \f@... macros 284
2019-07-09 ltfssini.dtx v3.1c
General: Explicitly set some defaults 388
2019-08-22 ltxref.dtx v1.1l
\labelformat: Commanded moved from varioref.sty 458
\Ref: Commanded moved from varioref.sty 458
\refstepcounter: Allow \p@... to have an argument 458
2019-08-27 fontdef.dtx v3.0c
General: Various commands made robust throughout the file 401
2019-08-27 ltboxes.dtx v1.3b
General: Various commands made robust 512
2019-08-27 ltclass.dtx v1.3b
\endfilecontents: Make various commands robust 662
2019-08-27 ltdefns.dtx v1.5f
General: Make various commands robust 98
\MakeRobust: Make the assignments global as we may need to apply them inside a group 82
2019-08-27 ltfilehook.dtx v1.2b
\unqu@tefilef@und: Make command robust 689
2019-08-27 ltfiles.dtx v1.2b
\IfFileExists: Make command robust 206
2019-08-27 ltfssbas.dtx v3.2d
General: Make various commands robust 268

2019-08-27 ltfssdcl.dtx v3.0s
\DeclareMathAccent: Make math accents robust 355
\set@mathdelimiter: Make math delimiters robust 362
2019-08-27 ltfssini.dtx v3.1d
General: Make various commands robust 367
2019-08-27 ltidxglo.dtx v1.1f
General: Make \index and \gloassary robust 622
2019-08-27 ltlength.dtx v1.1d
General: Make various command robust 266
2019-08-27 ltlogos.dtx v1.1j
\TeX: Make \TeX command robust . 192
2019-08-27 ltmath.dtx v1.2c
General: Make various commands robust 481
2019-08-27 ltmiscen.dtx v1.1p
General: Make various commands robust 460
\begin: Make command robust . . . 467
\end: Make command robust 470
2019-08-27 ltoutput.dtx v1.4e
\@begindvibox: Make \AtBeginDvi robust 734
2019-08-27 ltpage.dtx v1.0l
General: Make various commands robust 629
2019-08-27 ltpictur.dtx v1.1m
General: Make various commands robust 552
Remove several unnecessary \gdef definitions 552
2019-08-27 ltsect.dtx v1.1d
General: Make various commands robust 588
2019-08-27 ltspace.dtx v1.3j
General: Make various commands robust 173
2019-08-27 lttab.dtx v1.1q
General: Make various commands robust 527
Remove several unnecessary \gdef definitions 527
2019-08-30 lterror.dtx v1.2q
\conditionally@traceoff: Macro added 170
\conditionally@traceon: Macro added 170
2019-09-09 ltfssdcl.dtx v3.0s
\DeclareMathSymbol: Allow definition if the math symbol was a command already robust 357
2019-09-11 ltclass.dtx v1.3c
\endfilecontents: Support optional argument for filecontents 662
2019-09-14 ltfinal.dtx v2.1h
\@uclclist: Expand UTF8 chrs when case changing (github/177) 809
2019-09-16 ltxref.dtx v1.1m
General: Correctly revert the \p@... change 458
2019-09-21 fontdef.dtx v3.0d
General: Distangle alias (gh/184) 403, 407
2019-10-02 ltexpl.dtx v0.0
General: Initial version 65
2019-10-02 ltfinal.dtx v2.2
General: Load ltexpl 810
2019-10-02 ltluatex.dtx v1.1k
General: linebreak_ filter is exclusive 56
mlist_ to_ hlist is exclusive 56
process_ rule is exclusive 56
2019-10-07 lttab.dtx v1.1q
\extracolsep: This needs to expand 539
2019-10-11 ltfiles.dtx v1.2c
\set@curr@file: Remove one brace group 205
2019-10-11 ltfsstrc.dtx v3.0l
\@font@aliasinfo: Added 'alias' size function 333
2019-10-18 ltclass.dtx v1.3d
\load@onefilewithoptions: Initialize \...-h@@k only when loading the package or class (gh/198) 654
2019-10-22 ltluatex.dtx v1.1j
General: page_ objnum_ provider and process_ pdf_ image_ content classified data 56
2019-10-25 ltmiscen.dtx v1.1q
\add@percent@to@temptokena: Allow unbalanced conditionals in #1 (gh/202) 465
2019-10-26 ltfiles.dtx v1.2d
\@iffileonpath: quote on openin . . 207
\IfFileExists: dont quote name . . 206
\IfFileExists@: quote on openin . . 206
\set@curr@file: remove quotes . . . 205
2019-11-01 ltdirchk.dtx v1.3a
\filename@parse: take last . not first 12
2019-11-02 ltmiscen.dtx v1.1s
\@centercr: Make \@centercr robust (gh/203) 472
2019-11-02 ltspace.dtx v1.3k
\@normalcr: Make also \@normalcr robust 177

2019-11-09 ltfiles.dtx v1.2e
\set@curr@file: expand and \string before removing quotes 205
2019-11-10 ltmiscen.dtx v1.1r
\add@percent@to@temptokena: fix to special comment catcodes (gh/202) 465
2019-11-11 ltfiles.dtx v1.2f
\@iffileonpath: make \@filef@und match quoting used on \openin . 207
2019-11-22 ltoutenc.dtx v2.0l
General: Avoid spurious if fontenc selects LY1 as default encoding (gh/199) 256
2019-11-29 ltclass.dtx v1.3e
\@pr@videpackage: Protect package info text (gh/52) 644
2019-12-17 fontdef.dtx v3.0e
\mddefault: Set \bfdefault to "b" 395
\shapedefault: Set \shapedefault explicitly to "n" 396
\updefault: Set \updefault to "up" 395
2019-12-17 ltfntcmd.dtx v3.4c
textssc: Macro added 417
2019-12-17 ltfssbas.dtx v3.2e
\usefont: Don't call \fontseries or \fontshape 277
2019-12-17 ltfssini.dtx v3.1e
General: Provide custom series settings a la mweights 368
\DeclareEmphSequence: Provide \emph sequences 383
2019-12-18 ltoutenc.dtx v2.0m
General: Don't fake \textcompwordmark; take default from T1 instead 227
\add@accent: Avoid code that breaks \accent 220
2019-12-21 fontdef.dtx v3.0e
General: Distangle alias (gh/184) 401–404
2020-01-05 ltclass.dtx v1.3f
\endfilecontents: Support more write streams in LuaTeX gh/238 662
2020-01-11 ltfssini.dtx v3.1f
\rmfamily: Streamlined implementation with hook 378
\ttfamily: Streamlined implementation with hook 378
2020-01-20 ltfssdcl.dtx v3.0t
\set@mathdelimiter: fix for gh/251 362
2020-01-20 ltoutenc.dtx v2.0n
General: fix for gh/251 228
2020-01-22 lttextcomp.dtx v1.0b
\tc@subst: The overall default is \textcompsubstdefault not \substdefault 426
2020-01-25 fontdef.dtx v3.0f
General: Load t1enc.def last (gh/255) 392
2020-01-25 ltoutenc.dtx v2.0m
General: Load each encoding file only once (gh/255) 256
2020-01-27 ltclass.dtx v1.3g
\endfilecontents: Fix typo in error message 664
2020-01-28 ltclass.dtx v1.3h
\endfilecontents: Allow spaces in option string and display only unknown options not the whole option list (gh/256) 662
2020-01-31 ltvers.dtx v1.1e
General: Allow for upcoming format as pre-release 0 35
2020-02-02 ltluatex.dtx v1.1l
General: Add reverselist callback type 58
glyph_ info added 57
page_ order_ index added 56
post_ linebreak_ filter is reverselist 56
create_callback: Provide proper fallbacks for user-defined callbacks without user-provided default handler 60
2020-02-05 ltfssini.dtx v3.1g
\DeclareFontSeriesDefault: Clarrified error text 369
Corrected misspelled csname (gh/264) 369
2020-02-05 lttextcomp.dtx v2.0n
General: Changed the package default to info (gh/262) 442
Ensure we are on a new format (gh/260) 442
2020-02-07 ltfssini.dtx v3.1h
\symbol: XeTeX-specific version to avoid bug in maths mode. 386
2020-02-10 ltfssaxes.dtx v1.0c
\fontseries: Switch \if@forced@series added 303
\fontseriesforce: Switch \if@forced@series added 303
\if@forced@series: Switch \if@forced@series added 303
2020-02-10 ltfssini.dtx v3.1h
\@defaultfamilyhook: Add \@defaultfamilyhook to \normalfont (gh/269) 378

\reset@font: Add \@defaultfamilyhook to \normalfont (gh/269) 387
2020-02-10 lttextcomp.dtx v1.0c
General: Use \@tabaccludge for tabbing where necessary (gh/271) 429
2020-02-11 fontdef.dtx v3.0g
General: Provide value for \@fontenc@load@list (gh/273) . 392
2020-02-11 ltfssini.dtx v3.1h
General: Provide default value for \@fontenc@load@list (gh/273) . 388
2020-02-11 ltoutenc.dtx v2.0o
General: Update \@fontenc@load@list with option list (gh/273) 257
2020-02-14 ltpictur.dtx v1.1n
\linethickness: Suppress spaces following the declaration (gh/274) 557
2020-02-18 ltfssini.dtx v3.1i
\bfseries: Make the \ifx selection outside of \fontseries argument so that it is not done several times 375
\mdseries: Make the \ifx selection outside of \fontseries argument so that it is not done several times 376
\prepare@family@series@update: No series auto-update when forced (gh/277) 370
Recognize current family if it is not a "meta" family and auto-update series using \bfdefault (gh/277) 371
2020-02-18 ltmath.dtx v1.2d
\mathindent: Make \mathindent a skip register to match amsmath (gh/252) 493
equation: Separate formula and eqn number by at least a space in fleqn option 494
2020-02-20 ltclass.dtx v1.3j
\endfilecontents: Fix missing quotes around file name (gh/284) 664
2020-02-24 ltfssbas.dtx v3.2f
\DeclareFontShape@: Drop surplus "m" in series when defining fontshape (gh/289) 269
2020-02-25 ltfssini.dtx v3.1j
\bf@def@ult: Drop surplus "m" from \bfdef@ult and \mddef@ult (gh/291) 374
\prepare@family@series@update: Drop surplus "m" from \target@series@value (gh/291) 371
\update@series@target@value: Drop surplus "m" from \reserved@d (gh/291) 372
2020-02-27 ltdefns.dtx v1.5g
\@gobblethree: Macro added 79
2020-02-27 ltfssaxes.dtx v1.0d
\series@maybe@drop@one@m: Drop "m" in certain values from a fixed list (gh/293) 305
\set@target@series: Drop "m" only in a specific set of values (gh/293) 305
2020-02-27 ltfssbas.dtx v3.2g
\DeclareFontShape@: Only "m" if the series value is a member of a fixed list and issue warning if doing it (gh/293) 269
2020-03-02 ltexpl.dtx v1.0a
General: Don't load expl3 if already in the format (gh/295) 65
2020-03-05 ltexpl.dtx v1.1
General: Load xparse.ltx if \NewDocumentCommand is not defined by expl3.ltx 65
2020-03-06 ltboxes.dtx v1.3c
\clap: Macro \clap added 526
2020-03-07 ltluatex.dtx v1.1m
remove_from_callback: Do not call callback.register for user-defined callbacks 62
2020-03-07 ltmath.dtx v1.2e
\negthickspace: Add amsmath math/text spacing commands to the kernel (gh/303) 486
2020-03-07 ltspace.dtx v1.3l
General: Moved \thinspace, \negthinspace and \, to ltmath.dtx (gh/303) 173
2020-03-19 fontdef.dtx v3.0h
General: Support legacy use of \bfdefault and \mddefault (gh/306) 395
2020-03-19 ltfssdcl.dtx v3.0u
\document@select@group: fix for (gnats/3357) 345
2020-03-19 ltfssini.dtx v3.1k
\bfseries: Support legacy use of \bfdefault and \mddefault (gh/306) 375
\DeclareFontSeriesDefault: Support legacy use of \bfdefault and \mddefault (gh/306) 369
\mdseries: Support legacy use of \bfdefault and \mddefault (gh/306) 376

2020-04-06 ltfssini.dtx v3.1m
\bf@def@ult: Hook added (gh/306) 375
\bfseries: Hook added (gh/306) . . 375
\mdseries: Hook added (gh/306) . . 376
2020-04-07 ltclass.dtx v1.3k
\IfFormatAtLeastTF: Macro added; also in rollback (gh/168) 641
\load@onefile@withoptions: Use different method to ignore unprocessed options (gh/22) . . . 658
\ProcessOptions*: Use different method to ignore unprocessed options (gh/22) 649
\RequirePackageWithOptions: Use different method to ignore unprocessed options (gh/22) . . . 651
2020-04-09 ltfloat.dtx v1.2d
\@textsubscript: Set non-zero baseline (gh/249) 619
\textsubscript: Set non-zero baseline (gh/249) 619
2020-04-13 ltfssdcl.dtx v3.0v
\process@table: Small update for speed. 347
2020-04-13 ltfssini.dtx v3.1n
\init@series@setup: Handling \seriesdefault changes (gh/315) 373
\seriesdefault@kernel: Handling \seriesdefault changes (gh/315) 389
2020-04-21 ltmath.dtx v1.2f
\@yeqncr: Support calc syntax (gh/152) 491
2020-04-21 ltmiscen.dtx v1.1t
\@icentercr: Support calc syntax (gh/152) 472
2020-04-21 ltpictur.dtx v1.1o
\@istackcr: Support calc syntax (gh/152) 557
2020-04-21 ltspace.dtx v1.3m
\@hspace: Support calc syntax (gh/152) 190
\@newline: Support calc syntax (gh/152) 178
\@vspace@calcify: Support calc syntax (gh/152) 178
\@vspacer: Support calc syntax (gh/152) 187
\addvspace: Support calc syntax (gh/152) 184
2020-04-21 lttab.dtx v1.1r
\@itabcr: Support calc syntax (gh/152) 533
\@yargarraycr: Support calc syntax (gh/152) 541
2020-04-22 ltmiscen.dtx v1.1u
\@@sverb: Drop spaces before \verb delimiter (gh/327) 478
2020-04-22 ltoutenc.dtx v2.0p
General: y unicode value in tuenc.def 213
2020-04-29 lttextcomp.dtx v1.0d
General: Make all captial accents text commands for hyperref (gh/332) 429
2020-05-02 ltfiles.dtx v1.2g
\@include: Support spaces in filenames by enclosing the names of .aux-files in quotes (gh/217) . 203
\includeonly: Get rid of leading and trailing spaces from the filename (gh/217) 201
Improved support for spaces in filenames (gh/217) 201
Pass the filename to \@include by value instead of by reference (gh/217) 201
2020-05-05 ltxref.dtx v1.1n
\refstepcounter: record the counter name in \@currentcounter 458
2020-05-06 ltspace.dtx v1.3n
General: Made softhyphen active in TU engines 191
2020-05-09 ltdefns.dtx v1.5j
\@if@DeclareRobustCommand: Added \DeclareCommandCopy (gh/239) . . 90
\DeclareCommandCopy: Added \DeclareCommandCopy (gh/239) . . 88
2020-05-11 ltdefns.dtx v1.5j
\@dischyph: Do not overwrite \- under LuaTEX 97
2020-05-15 ltdefns.dtx v1.5g
\typeout: Allow \par in the argument (gh/335) 70
2020-05-19 ltfssaxes.dtx v1.0e
\series@maybe@drop@one@m: Need to use \edef (gh/336) 305
2020-05-19 ltfssini.dtx v3.2a
\IfFontSeriesContextTF: Macros added (gh/335) 380
2020-05-31 ltmiscen.dtx v1.1u
\centering: Added \finalhyphendemerits setting (gh/247) 473
\raggedleft: Added \finalhyphendemerits setting (gh/247) 473
\raggedright: Added \finalhyphendemerits setting (gh/247) 473

2020-06-04 ltexpl.dtx v1.2c
General: Define a local version of some LaTeX 2ε basic macros to support package loading 65
2020-06-04 ltfinal.dtx v2.2a
General: Load ltexpl in `ltdefns` . . . 810
2020-06-05 ltclass.dtx v1.3l
`\@currnamestack`: Added `\@expl@pop@filename@@` 640
Added `\@expl@push@filename@@` and `\@expl@push@filename@aux@@` . . 638
2020-06-05 ltfiles.dtx v1.2h
`\document`: Added hook to load l3backend code 196
2020-06-10 ltluatex.dtx v1.1n
General: Define `\@gobble`/`\@firstofone` even for LaTeX to allow early loading. 42
2020-07-04 ltoutenc.dtx v2.0q
General: Implement `\remove@tlig` in LuaTeXwithout font reloading . . 245
2020-07-08 ltexpl.dtx v1.2d
General: Add a last-minute hook for expl3 65
2020-07-08 ltfinal.dtx v2.2b
General: Add a last-minute hook for expl3 802
2020-07-27 ltmath.dtx v1.2g
`\cases`: Don't make the command `\long` (gh/354) 485
`\matrix`: Don't make the command `\long` (gh/354) 485
`\pmatrix`: Don't make the command `\long` (gh/354) 485
2020-07-27 ltoutenc.dtx v2.0r
`\@use@text@encoding`: Don't make the command `\long` (gh/354) . 222, 223
2020-07-27 ltpage.dtx v1.0m
`\markright`: Don't make the command `\long` (gh/354) 630
2020-07-27 ltpictur.dtx v1.1p
`\linethickness`: Don't make the command `\long` (gh/354) 557
2020-07-27 ltsect.dtx v1.1e
`\author`: Don't make the command `\long` (gh/354) 588
`\date`: Don't make the command `\long` (gh/354) 588
2020-08-01 ltluatex.dtx v1.1p
General: new_ graf is `exclusive` . . . 56
2020-08-02 ltluatex.dtx v1.1q
`\newattribute`: Move reset to 0 inside conditional 44
`\newluabytecode`: Move reset to 0 inside conditional 47
`\newluachunkname`: Move reset to 0 inside conditional 47
`\newluafunction`: Move reset to 0 inside conditional 46
`\newwhatsit`: Move reset to 0 inside conditional 47
2020-08-08 ltclass.dtx v1.3m
`\endfilecontents`: define `\q@curr@file` directly as the quotes have already been removed (gh/220) 663
2020-08-10 ltluatex.dtx v1.1r
General: Load ltluatex Lua module during format building 47
2020-08-15 ltpictur.dtx v1.2a
`\@defaultunitsset`: Macro added . 554
2020-08-19 ltdefns.dtx v1.5k
`\@carcube`: Made `\long` for `\NewCommandCopy` 73
`\robust@command@act`: Made `\robust@command@act` (was `\declare@command@copy`) more generic 86
`\ShowCommand`: Added `\ShowCommand` (gh/373) 89
2020-08-19 ltexpl.dtx v1.2e
General: Add `\@expl@cs@`⟨*thing*⟩`@spec@@N` for `\ShowCommand` (gh/373) 68
Add `\@expl@cs@to@str@@N` and `\@expl@str@if@eq@@nnTF` for `\NewCommandCopy` (gh/239) 68
2020-08-20 ltplain.dtx v2.3d
`\alloc@`: Define `\alloc@` in terms of `\e@alloc` 21
2020-08-21 ltclass.dtx v1.3o
General: Integration of new hook management interface 633
2020-08-21 ltdefns.dtx v1.5l
General: Integration of new hook management interface 70
2020-08-21 ltdefns.dtx v1.5m
`\MakeRobust`: Make `\MakeRobust` produce the same command structure as `\DeclareRobustCommand` 82
2020-08-21 ltexpl.dtx v1.2d
General: Dropped unused command . 65
2020-08-21 ltfiles.dtx v1.2i
General: Integration of new hook management interface 193

2020-08-21 ltfinal.dtx v2.2i
General: Integration of new hook management interface 797
2020-08-21 ltfssaxes.dtx v1.0g
General: Integration of new hook management interface 311
2020-08-21 ltfssini.dtx v3.2b
`\bf@def@ult`: Integration of new hook management interface 375
`\bfseries`: Integration of new hook management interface 375
`mdseries/defaults`: Integration of new hook management interface 378
`\mdseries`: Integration of new hook management interface 376
`\reset@font`: Integration of new hook management interface 387
`\rmfamily`: Integration of new hook management interface 378
`\ttfamily`: Integration of new hook management interface 378
2020-08-21 ltmiscen.dtx v1.1v
General: Integration of new hook management interface 460
2020-08-21 ltoutput.dtx v1.4f
`\@begindvibox`: Integration of new hook management interface 734
2020-08-23 ltxref.dtx v1.1o
`\refstepcounter`: add default definition of `\@currentcounter` . 458
2020-09-06 ltclass.dtx v1.3q
`\load@onefilewithoptions`: Save `\@currpkg@reqd` so that we don't lose track of package substitutions. 656
2020-09-06 ltdefns.dtx v1.5n
`\char@if@alph`: Macro added 96
2020-09-06 ltexpl.dtx v1.2f
General: Add `\@expl@str@map@function@@NN and \@expl@char@generate@@nn` for `\string@makeletter` (gh/386) 68
2020-09-09 ltshipout.dtx v1.0b
`\__shipout_picture_overlay:n`: Prevent overfull box warnings (gh/387) 713
2020-09-26 ltfinal.dtx v2.2j
General: Load first aid file if existing 811
2020-09-30 ltfssini.dtx v3.2d
`\bfseries`: `\bfdefault@previous` not `\bfseries@previous` (gh/395) . . 375
`\mdseries`: `\mddefault@previous` not `\mdseries@previous` (gh/395) . . 376
2020-10-01 ltclass.dtx v1.3r
`\@pr@videpackage`: Allow for package subsitution 644
2020-10-01 ltsect.dtx v1.1e
`\addcontentsline`: add a fourth argument for better hyperref compability 597
2020-10-04 ltfiles.dtx v1.2j
`\@include`: Quotes around the aux file name removed, they are not needed and upset BibTeX (gh/400) 203
2020-10-04 lthooks.dtx v1.0d
General: Definition `\AddToHookNext` was supposed to be for `\AddToHook` vize versa (gh/401) 153
2020-10-08 ltclass.dtx v1.3s
`\@currnamestack`: Added missing 2020/02/02 `\IncludeInRelease` . 638
2020-10-11 ltclass.dtx v1.3t
`\load@onefilewithoptions`: Restore `\@currpkg@reqd` after finished loading a package file (gh/408). . 656
2020-10-18 ltclass.dtx v1.3t
`\PassOptionsToClass`: Drop path from `\input@path` (gh/414). . . . 646
2020-10-23 ltmiscen.dtx v1.1w
`\enddocument`: Make `enddocument/afteraux` one-time 462
2020-11-09 ltclass.dtx v1.3u
`\pkgcls@rollbackdate@error`: Change help text because the package may have existed then — there is just no rollback data (gh/423). 676
2020-11-09 ltmath.dtx v1.2h
`\negthickspace`: `\negmedspace` and `\negthickspace` have been only in amsmath, so we need to undefine for rollback (gh/423) 487
2020-11-20 ltclass.dtx v1.3u
`\@currpath`: Macro added 638
`\@kernel@currpathstack`: Macro added 640
`\load@onefile@withoptions`: Copy option list to the requested package. 657
`\PassOptionsToClass`: Copy option list to the requested package. . . . 646
`\ProvidesPackage`: Use string comparison instead of `\ifx` 644
2020-11-20 ltfilehook.dtx v1.0d
`\unqu@tefilef@und`: Move loading to `\@input@file@exists@with@hooks`

and expand `\@filef@und` to avoid getting the wrong file name in the case of a substitution. 688
2020-11-24 ltexpl.dtx v1.2g
General: Load xparse in latexrelease if needed (gh/434) 67
Support for roll forward (gh/434) 66, 68
2020-11-24 ltfilehook.dtx v1.0d
General: Support for roll forward (gh/434) 686
2020-11-24 lthooks.dtx v1.0f
`\__hook_curr_name_pop:`: Support for roll forward (gh/434) 151
2020-11-24 ltshipout.dtx v1.0d
General: Support for roll forward (gh/434) 718
`\AtBeginDvi`: Support for roll forward (gh/434) 717
2020-11-25 ltdefns.dtx v1.5o
`\@carcube`: Added missing latexrelease entry 73
2020-12-02 ltluatex.dtx v1.1s
General: Fix return value of list callbacks 58

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

\! b364, b366, E201, U421
\" . o226, o378, o416, o454, o465, o573, o605, o632, o640, o646, o650, o656, o660, o666, o672, o679, o680, o686, o690, o740, o786, o1210, o1228, o1234, o1238, o1244, o1248, o1254, o1260, o1267, o1268, o1274, o1278, o1280, o1387, r387, A185, A202, U422
\# a62, a75, b6, b14, b431, f696, r374, U405
\$. a74, b4, b13, f695, o307, o441, o448, o562, o798, o805, A547, A554, U406
\% a75, a105, a107, a127, b14, b429, f696, o488, o490, r376, A556, A670, D124, P1277, P1278, U407
\& . . . a74, b5, b13, b430, f695, P343, U408
\' . b451, o227, o379, o418, o452, o462, o575, o585, o591, o593, o596, o598, o606, o612, o618, o620, o623, o625, o633, o637, o644, o648, o653, o658, o661, o663, o670, o675, o676, o683, o688, o691, o741, o788, o807, o809, o810, o811, o814, o816, o817, o818, o820, o821, o1204, o1225, o1232, o1236, o1241, o1246, o1249, o1251, o1258, o1263, o1264, o1271, o1276, o1279, o1287, o1288, o1335, o1336, o1341, o1342, o1353, o1354, o1359, o1360, o1388, o1389, o1413, o1414, r386, w588, x253, A180, D562, E241, G271, G292, H72, S584, U423
\(. E258, E332
\) b451, E258, E333
* . r379, E238, P958, P1089, P1203, P1279
\+ H72
\, ... b365, b367, x506, D562, E7, E8, E40, E154, E156, E159, E184, E207, E208, E224
\- . . *98*, *191*, b333, f24, f730, l416, l478, o413, o414, o568, o782, o783, r381, D562, G270, G291, H72, U157, U197
\. b364, b366, n47, n116, n174, o228, o380, o449, o450, o471, o581, o582, o608, o609, o635, o742, o812, o819, o1209, o1291, o1292, o1301, o1302, o1311, o1312, o1329, o1390, o1391, o1421, o1422, r380, A186, A203
\..default *295*
\/ a97, f25, r328, r382, P342
\: b365, b367, f690, f691, E201, E239
\; *486*, b365, b367, x500, E185, E201
\< o569, o733, r377, D562, H71, H109
\= o229, o381, o470, o743, o1207, o1281, o1282, o1297, o1298, o1320, o1321, o1322, o1347, o1348, o1373, o1374, o1403, o1404, o1419, o1420, o1427, o1428, w588, A189, G271, G292, H71
\> o566, o734, r378, D562, E213, E228, E229, E239, H71
\? b364, b366, U423
@@ commands:
  \@@_set_curr_file:nNN Q322
  \@@_set_curr_file_assign:nnnNN Q322
\@@par *171*
\@TeXversion *1*, *6*
@botlist commands:
  \@botlist: S1029, S1123, S1282
@botnum commands:
  \@botnum: S1003
\@car *72*
\@cdr *72*
@citeb commands:
  \@citeb: N16, N41
@colht commands:
  \@colht: S541, S555
@colnum commands:
  \@colnum: S1188, S1356
\@cons *72*
\@currdir *1*, *6*
@currname commands:
  \@currname: n614
\@dblarg *71*
@dbldeferlist commands:
  \@dbldeferlist: S1755, S1797
@dbltopnum commands:
  \@dbltopnum: S1654, S1781
@deferlist commands:
  \@deferlist: S1114, S1211, S1272, S1380, S1424, S1453, S1511, S1543, S1629, S1669
\@ifclasslater *636*
\@ifclassloaded *636*
\@ifclasswith *636*
\@ifdefinable *71*
\@ifnextchar *71*
\@ifpackagelater *636*

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@ifpackageloaded *636*
\@ifpackagewith *636*
\@ifstar *71*
\@ifundefined *71*
\@missingfileerror *637*
\@namedef *71*
\@nameuse *71*
\@restorepar *171*
\@setpar *171*
@topnum commands:
\@topnum: S1049
\[r383, w40, w51, E276, E334, E431, U424
\\ *173*
\\ a45, a46, a74, a247, a248, a249, a250, a253, a260, a261, a262, a263, a266, a273, a274, a275, a276, a279, a286, a292, a293, a297, a299, a300, a304, a309, a310, a313, a319, b13, d264, d416, f219, f276, f437, f525, f543, f695, g575, g586, g874, g895, j227, l52, l480, n607, n622, o557, r371, x251, D344, D349, D354, D364, D368, D372, D396, E356, E491, G303, G411, G413, H73, H170, H180, H194, U409, I139
\{ a3, a7, a74, b2, b13, j22, o308, o559, r372, x249, D395, E59, E154, U412
\} a8, a74, b3, b13, j21, o309, o560, r373, x250, D395, E59, U413
\⟨*addto-cmd*⟩ *101*
\⟨*env*⟩ *116*
\⟨*filename*⟩ *692*
⟨*name*⟩ internal commands:
__hook next_⟨*name*⟩ *122*
\] . . . b451, r384, E276, E335, E455, U425
\^ a63, a72, a75, a119, a330, b7, b9, b11, b14, b370, b371, b385, b386, f20, f696, l480, l482, l484, o231, o286, o382, o453, o463, o555, o638, o645, o649, o654, o659, o664, o671, o677, o678, o684, o689, o744, o1205, o1222, o1226, o1233, o1237, o1242, o1247, o1252, o1259, o1265, o1266, o1272, o1277, o1289, o1290, o1307, o1308, o1315, o1316, o1330, o1331, o1332, o1361, o1362, o1383, o1384, o1385, o1386, r369, r370, r375, A184, A201, D116, D125, P959, P960, P961, P1043, P1046, P1049, P1090, P1091, P1092, P1174, P1177, P1180, P1204, P1205, P1206, P1264, P1267, P1270, U188, U189, U190, U191, U192, U193, U194, U195, U196, U410, U416, U417, U418, U419, U453, U454, U455, U456, U457, U458, U459, U460, U461
_ a75, b8, b14, f696, o314, x254, E256, E257, U411
\‘ . o232, o383, o417, o451, o461, o574, o636, o643, o647, o652, o657, o662, o669, o673, o674, o682, o687, o745, o787, o1203, o1224, o1231, o1235, o1240, o1245, o1250, o1257, o1261, o1262, o1270, o1275, r385, w588, A187, D562, G271, G292, H72, U426
\| o558, p135, p146, x573, x574, U427
\~ a75, b10, b14, f696, j20, l422, o239, o287, o384, o464, o556, o639, o651, o655, o665, o681, o685, o746, o1206, o1223, o1227, o1239, o1243, o1253, o1269, o1273, o1317, o1318, o1319, o1371, o1372, A193, A211, D493, D508, D523, D566, U414
\␣ a74, a91, b13, b370, b388, f695, j19, j20, j21, j22, j25, l421, r368, r618, r654, r679, x252, D392, D393, D498, D513, J36, J38, N17, P336, U404

A

\A U185, U429, U450
\a o223, H1, U176, U430, U441
\AA b376, o240, o425, o523
\aa b376, o245, o419, o533
\abovedisplayshortskip b351, E499
\abovedisplayskip b350, E492, E494, E496, E497, E498, E499
\accent . . *305*, o73, o393, o420, o476, o757
\active a64, a119, a330, b10, b11, b385, b386, b388, D392, D393, D492, D498, D507, D513, D522, D564, E241, E256, P959, P960, P961, P1043, P1046, P1049, P1090, P1091, P1092, P1174, P1177, P1180, P1204, P1205, P1206, P1264, P1267, P1270, S584
\acute x516
add commands:
add_to_callback d704
\add_to_callback *41*
\addcontentsline *464*, K70, K80, K159, L16
\addpenalty *185*, l270, F124, F170, F175, K50, S338, S1153, S1319
\addtocontents *465*, K164, K171, K177, K180

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\addtocounter *258*
\addtocounter p6, p18
\AddToHook *100*, *101*, *102*, *102*, *103*, *106*, *110*, *113*, *114*, *115*, *704*, *704*, *704*, *704*, g901, g972, D37, D38, D289, D290, D291, D292, P864, P865, P866, Q488, Q490, Q500, Q501, Q502, Q503, R114, R394, R417, R443
\AddToHookNext *103*, *103*, *106*, *110*, *144*, *698*, *702*, *704*, *704*, g903, g970, Q489, R418, R444
\addtolength *266*
\addtolength q16, E494, E496
\addtoversion u20, u139
\addvspace l233, D322, F124, F171, F172, F176, F224, K50
\adjdemerits b330
\AE o241, o398, o524, o762, o1102, o1403, U482
\ae o246, o401, o534, o766, o1108, o1404, U482
\afterassignment *83*, b401, b404, f256, f262, f305, o212, o220, r299, E186
\AfterEndEnvironment *116*, D285
\aftergroup r91, r313, t156, t222, v114, v121, v129, z64, D496, D511, D526, G129, S604, S605, S662, S663
\AfterLastShipout Q500
\afterpreamble *197*
\aleph x308
\allocationnumber b37, b57, b69, b71, b143, b144, b145, b195, b196, b237, b238, b239, b252, b253, b254, b271, b277, b283, b284, b297, b298, b299, d52, d53, d54, d91, d205, H4, H9, U44, U45, U46
\allowbreak b408, f772, f773, f792, f794, E40
\Alph *258*
\Alph *703*, p106
\alph *258*
\alph p105
\alpha x268
\amalg x379
\AmSfont *303*
\and *588*
\and K14, K27
\angle x337
\approx x422
\arabic *258*
\arabic *703*, p102, J33, R274
\arccos E13
\arcsin E10
\arctan E16
\arg E26
array (environment) H168
\array H168
\arraycolsep E359, E360, E504, E505, H258, H338
\arrayrulewidth . . H324, H338, H346, H347, H359, H363, H366, H376, H378
\arraystretch H186, H187, H342
\Arrowvert x569
\arrowvert x567
\asciispace D452, D454, D457, D460, D461, D480
\ast x232, x395
\asymp x449
\AtBeginDocumement *115*
\AtBeginDocument *101*, *114*, *115*, *115*, *116*, *117*, *309*, n127, n182, s619, N34, N48, P855
\atbegindocumenthook *197*
\AtBeginDvi *115*, *702*, *704*, *709*, *709*, *717*, R327, R361, S86
\AtBeginEnvironment *116*, D285
\AtBeginShipout *704*, R366, R417
\AtBeginShipoutAddToBox *704*, R377, R422
\AtBeginShipoutAddToBoxForeground *704*, R378, R422
\AtBeginShipoutBox *704*, R415
\AtBeginShipoutDiscard . *704*, R375, R421
\AtBeginShipoutFirst . . . *704*, R369, R419
\AtBeginShipoutInit *704*, R416
\AtBeginShipoutNext *704*, R367, R417
\AtBeginShipoutUpperLeft *704*, *704*, R379, R422
\AtBeginShipoutUpperLeftForeground *704*, *704*, R380, R422
\AtEndAfterFileList Q502
\AtEndDocument *115*, *117*, D70, P855, Q506
\AtEndDvi *702*, *705*, R384, R389
\AtEndEnvironment *116*, D285
\AtEndOfClass *681*, E430, P855
\AtEndOfPackage *681*, P444, P463, P530, P541, P855
\AtEndPreamble *117*
\AtNextShipout *704*, R443
\atopwithdelims E57, E58, E59
\attribute *38*, d79
\attributedef d79, d215
\attributezero d215
\AtVeryEndDocument Q501
\AtVeryVeryEnd Q503
\author *588*
\author K8, K24, K32

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

B

\b o233, o389, o472, o753, o1214
\backslash x251, x590
\bar x520
\baselineskip . . . . . .
. . . . b369, b399, b435, t140, t141, t142, t144, t145, x510, E158, E159, E178, E184, E188, G280, G299, H198, R146, R197, S242, S273, S622, S637, S681, S696, I136, I314, I373
\baselinestretch r290, t118, t119, t138, t199
\batchmode *210*, e70, e76, n552, n583, n584, u106, U502, U523
\BeforeBeginEnvironment . . . . . .
. . . . . *116*, *116*, *116*, *116*, D285
\BeforeClearDocument Q504
\begin *82*, *100*, *115*, *116*, *116*, *117*, *130*, *195*, *469*, *469*, j197, j199, t7, x4, x102, y4, D156, D157, E435, E447, K14, K17, P551, R309, R455, T3, U387, U391, I461
\begingroup *195*, *220*, *469*
\belowdisplayshortskip b353, E498
\belowdisplayskip b352, E497
\beta x269
\bezier *552*
\bezier . . I682, I683, I805, I806, I821, I823
\bfdefault . . . *119*, *305*, *368*, *368*, *369*, *369*, *370*, *371*, *373*, *374*, *375*, *375*, *375*, *375*, *375*, w15, w194, w201, w202, w203, w204, w213, w274, w285, w326, x92, x104, x106, x114
\bfseries *119*, *368*, *368*, *371*, *380*, w13, w14, w189, w198, w268, w279, w280, w320, w324, w325, z19, C13, J36, J38, N20, R310
bfseries w252
bfseries/defaults w252
\bgroup b383
\bibcite N7, N9, N10
\bibdata N25, N29
\bibitem N3
\bibliography *625*
\bibliography N27
\bibliographystyle *625*
\bibliographystyle N32
\bibstyle N25, N37
\Big . x623, x626, x635, x637, E44, E45, E46
\big x624, x636, E41
\bigbreak b415, f774, f795
\bigcap x345
\bigcirc x392
\bigcup x346
\Bigg x630, x639, E50, E51, E52
\bigg x628, x638, E47, E48, E49
\Biggl E50
\biggl E47
\Biggm E51
\biggm E48
\Biggr E52
\biggr E49
\Bigl E44
\bigl E41
\Bigm E45
\bigm E42
\bigodot x353
\bigoplus x352
\bigotimes x351
\Bigr E46
\bigr E43
\bigskip b420, l400
\bigskipamount b419, l402, l403, L367
\bigsqcup x356
\bigtriangledown x361, x362
\bigtriangleup x360, x363
\biguplus x344
\bigvee x342
\bigwedge x343
\binoppenalty b321
\bmod E35
\boldmath m14, w463
bool commands:
\bool_gset_false:N g15, R15, R43, R51
\bool_gset_true:N g10, R10, R85, R268
\bool_if:NTF . . . . . .
. . . . g21, g664, g673, R21, R49, R289
\bool_lazy_and:nnTF g666, g833, R291
\bool_lazy_or:nnTF g259
\bool_new:N . . g6, g24, R6, R112, R127
\bool_set_false:N g656
\bool_set_true:N g653
\bool_while_do:nn g483
\bordermatrix E172
\bot x321
\botfigrule S742, S2325
\botmark O50, S647, S706
\bottomfraction L275, S2294
\bowtie x482
\Box w539
\box *497*
box commands:
\box_dp:N R118
\box_ht:N *706*, R117, R226
\box_if_empty:NTF R40, R58
\box_if_horizontal:NTF . . R158, R210
\box_if_vertical:NTF . . . R132, R183
\box_move_up:nn R172, R226
\box_new:N R23, R128

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\box_set_dp:Nn R145, R154, R171, R196, R207, R225, R256
\box_set_eq_drop:NN R54
\box_set_ht:Nn R144, R153, R170, R195, R206, R224, R255
\box_set_wd:Nn R143, R169, R194, R223
\box_use:N R88, R149, R174, R202, R227, R257
\box_wd:N R119, R220, R228
\l_shipout_box *700*, *701*, *706*, *706*, *710*, *711*, R23, R29, R40, R46, R54, R58, R69, R88, R131, R132, R139, R150, R153, R154, R158, R165, R175, R183, R190, R200, R206, R207, R210, R217, R219, R220, R226, R228
\boxmaxdepth b344, S486, S506, S546, S715, S724, S764, I475, I503, I533, I611, I628
\brace E59
\braceld x553, x557, x558, x560, x562
\bracelu x555, x559, x561
\bracerd x554, x559, x561
\braceru x556, x558, x562
\bracevert x608
\brack E58
\break b408, b413, f775, f796, l115
\breve x521
\brokenpenalty b326, r637
\buildrel x469, E149
\bullet x381

C

\c o234, o335, o337, o339, o341, o343, o345, o347, o349, o351, o372, o374, o392, o456, o475, o600, o602, o627, o629, o642, o668, o695, o698, o699, o700, o701, o702, o703, o704, o705, o706, o756, o1216, o1230, o1256, o1313, o1314, o1333, o1334, o1337, o1338, o1343, o1344, o1355, o1356, o1363, o1364, o1367, o1368, A183, A200
\cal w589
call commands:
call_callback d685
\call_callback *42*
callback commands:
callback_descriptions d815
callback.register d565
\callback_descriptions *42*
\cap x372
\capitalacute o833, A180, A197, A600, A852
\capitalbreve o840, A181, A198, A601, A859
\capitalcaron o839, A182, A199, A602, A858
\capitalcedilla o826, A183, A200, A603, A849
\capitalcircumflex o834, A184, A201, A604, A853
\capitaldieresis o836, A185, A202, A605, A855
\capitaldotaccent o842, A186, A203, A606, A861
\capitalgrave o832, A187, A204, A607, A851
\capitalhungarumlaut o837, A188, A205, A608, A856
\capitalmacron o841, A189, A206, A609, A860
\capitalnewtie *429*, o846, A195, A207, A610, A926, A927
\capitalogonek o829, A190, A208, A611, A850
\capitalring o838, A191, A209, A612, A857
\capitaltie o844, A192, A210, A613, A922, A923
\capitaltilde o835, A193, A211, A614, A854
\caption L4
\cases E153, E154, E164, E166
\catcode *465*
\catcodetable *38*, d89, d109
\catcodetable@atletter *39*
\catcodetable@initex *39*
\catcodetable@latex *39*
\catcodetable@string *39*
\cdot x394
\cdotp x502, x508
\cdots x508
center (environment) D337
\center D337
\centering D337, D342, D343, D361, D363, D378, D380
\centerline G428
\changes *68*
\char f737, f752, o391, o394, o427, o430, o441, o448, o474, o478, o483, o486, o488, o490, o727, o755, o758, o791, o798, o805, o828, o831, o879, o989, o1025, o1182, w476, w483, A547, A554, A556, A670, D452, D567, E238, I232, I282, I296, I304, I307, I448, I563, I568, I576, I580, I616, I617, I619, I632, I633, I636, I663
char commands:
\char_generate:nn e138
\chardef a64, a70, a71, b10, b16, b17, b18, b19, b20, b58, b64, b66, b73, b79, b82, b84, b94, b96, b97, b98, b99,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

b108, b114, b115, b128, b130, b194, b253, b257, b259, b283, b298, b429, b430, b431, d22, d26, d38, d47, d48, d89, d157, d216, h2, n54, n123, o18, r14, H4, H9, P965, P1096, P1213, U28, U30, U34, U53, U104, U105, U106, U107, U108, U109, U110
\charsubdef U251
\charzero d216
\check x522
\CheckCommand f187
\CheckEncodingSubset . *423*, A16, A82, A130, A131, A132, A176, A178, A286, A537, A741, A791, A847, A848, A916, A1033, A1036, A1050
\chi x288
\choose E57
\circ x391
\circle I450, I605, I807, I824
\citation N11, N19, N43
\cite *625*
\cite N12
\clap G432
class/after *681*
class/after/... . . . . *681*
class/before *681*
class/before/... . . . . *681*
\ClassError j84
\ClassInfo j84
\ClassWarning j84
\ClassWarningNoLine j84
\cleaders b449, x548, x551
\cleardoublepage S138
\ClearHookRule *105*, g949, g986
\clearpage *117*, *204*, *681*, *682*, *715*, *715*, n297, n324, n328, n344, n362, D16, D73, D154, S125, S138, S143, S200, S407, S410, S414, S455, S461, S2171, S2188
\cline H367
clist commands:
  \clist_gclear:N g482
  \clist_gput_left:Nn g426
  \clist_gput_right:Nn g428
  \clist_if_empty:NTF g683
  \clist_new:N g51
  \clist_use:Nn g685
\clubpenalty *195*, b323, n10, n27, n96, n153, r635, F128, F194, F196, K100, K106, K130, K135
\clubsuit x331
\colon x503
\columnsep n29, n98, n155, S81, S202
\columnseprule S82, S2226, S2260
\columnwidth n26, n29, n30, n32, n95, n98, n99, n101, n152, n155, n156, n159, G327, G354, L99, L168, L442, S80, S144, S145, S146, S201, S202, S203, S204, S205, S1843, S1845, S2224, S2228, S2256, S2262
\cong x457
\contentsline *597*, K164, K171, K177, K184
\coprod x341
\copyright o285, o316, w499
\cos E12
\cosh E14
\cot E18
\coth E19
\countdef a66, b37, b38, b39, b41, b51, b90, d75, d85, d174, d182, d190, d198, d217, B3, U61
\counterwithin *258*
\counterwithin p77, p97
\counterwithout *258*
\counterwithout p67, p94
\CountZero d217
\cr b379, o497, o503, o513, o519, A122, A126, A839, A843, E175, E179, E364, E410, E508, H192, H203, H210, H219, H224, H230, H377, I141, I143, I148, I154
\crcr b436, o327, o362, o363, o390, o394, o397, o473, o477, o481, o483, o486, o726, o754, o758, o761, o828, o831, o878, o1219, w501, x337, x338, x340, x459, x462, x466, x530, x531, x532, x533, x534, x535, x537, x538, x539, x540, x541, x543, A127, A844, E155, E157, E158, E159, E175, E177, E178, E179, E197, E198, H171, H172, I141
create commands:
  create_callback d660
\create_callback *42*
\CS *72*
cs commands:
  \cs:w *125*, g135, g389, g422, g474, g494, g497, g536, g537, g538, g545, g552, g774, g784, g814, g855, Q203, Q216, Q228, Q229
  \cs_argument_spec:N e135
  \cs_end: g135, g392, g422, g474, g494, g497, g527, g537, g538, g545, g552, g722, g774, g780, g793, g814, g855, Q203, Q216, Q228, Q229
  \cs_generate_variant:Nn g37, g38, g39, g190, g515, Q397, Q405

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\cs_gset:Npn Q233, R106
\cs_gset:Npx Q205
\cs_gset_eq:NN .. e132, e133, e134, e135, e136, e137, e138, g396, g409, g410, g768, R31, R105, R245, R302
\cs_gset_protected:Npx g20, R20
\cs_if_exist:NTF *694, 695*, Q92, Q374, Q378
\cs_if_exist_use:NTF g338, R237, R238, R242, R243
\cs_new:Npn g87, g96, g109, g119, g120, g122, g132, g138, g271, g273, g275, g323, g371, g387, g439, g440, g516, g517, g698, g778, g786, g794, g938, g939, Q30, Q38, Q44, Q57, Q94, Q99, Q104, Q110, Q124, Q225, Q231, Q347, Q349, Q351, Q354, Q358, Q372, Q388, Q398, Q509, R33, R39, R102, R113, R116, R130, R181, R234, R247, R261, R264, R274, R307, R431, R432, R433
\cs_new_eq:NN .. g7, g23, g36, g352, g358, g394, g591, g592, g593, g594, g595, g596, g793, g951, g952, g954, g956, g958, Q243, Q244, Q441, Q443, Q445, Q447, Q449, Q451, Q453, Q455, Q457, Q459, R7, R94, R271, R273, R326, R328, R329
\cs_new_protected:Npn *717*, g8, g13, g18, g41, g43, g56, g64, g66, g71, g73, g139, g148, g153, g161, g171, g173, g191, g204, g209, g214, g219, g228, g286, g288, g305, g307, g312, g328, g333, g347, g353, g359, g364, g366, g368, g369, g395, g412, g446, g518, g525, g534, g541, g548, g555, g563, g569, g580, g597, g604, g616, g621, g626, g628, g630, g700, g715, g720, g727, g743, g745, g754, g759, g767, g769, g791, g800, g809, g817, g827, g917, g923, g940, g941, g942, g943, Q49, Q61, Q69, Q78, Q200, Q213, Q328, Q333, Q356, Q474, R8, R13, R18, R267, R394
\cs_prefix_spec:N e134
\cs_replacement_spec:N e136
\cs_set:Npx Q203, Q216
\cs_set_eq:NN g425, g426, g427, g428, g618, g623, R24, R44, R56, R87, R91, R331, R333, R335, R337, R339
\cs_set_protected:Npn g702, g730, g907, R25, R327
\cs_to_str:N *125*, e132
\cs_undefine:N g370, Q218
cs\check@icr commands:
\cs_gset_eq:NN *68*
\csc E21
\csname *103, 372, 469, 691, 694*
\csname\endcsname *691, 692*
\cup x373
\CurrentFile *203, 656, 680, 680, 680, 680, 682, 685, 688, 689, 693, 693*, P698, Q3, Q66, Q83, Q155, Q167, Q169, Q313, Q317, Q482, Q483
\CurrentFilePath . *680, 680, 685, 688, 693*, Q3, Q66, Q82, Q154, Q314, Q317
\CurrentFilePathUsed *680*, Q3, Q65, Q80, Q152, Q311, Q314
\CurrentFileUsed *656, 680, 682, 688, 693*, P698, Q3, Q65, Q81, Q153, Q311, Q313, Q481, Q482
\CurrentOption o1496, o1499, o1504, o1516, P13, P395, P406, P407, P412, P420, P421, P424, P436, P441, P442, P455, P460, P461, P469, P471, P481, P482, P483, P491, P492, P493, P731, P789, P887, P888, P898
CurrentOption commands:
\CurrentOption: P405, P419, P435, P440, P454, P459, P480, P490, P897
\CYRA o1468
\cyra o1468, o1519
\CYRABHCH o1468
\cyrabhch o1468
\CYRABHCHDSC o1468
\cyrabhchdsc o1468
\CYRABHDZE o1469
\cyrabhdze o1468
\CYRABHHA o1469
\cyrabhha o1469
\CYRAE o1469
\cyrae o1469
\CYRB o1469
\cyrb o1469
\CYRBYUS o1470
\cyrbyus o1469
\CYRC o1470
\cyrc o1470
\CYRCH o1470
\cyrch o1470
\CYRCHLDSC o1470
\cyrchldsc o1470
\CYRCHRDSC o1471
\cyrchrdsc o1470
\CYRCHVCRS o1471
\cyrchvcrs o1471
\CYRD o1471

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\cyrd o1471
\CYRDELTA o1471
\cyrdelta o1471
\CYRDJE o1472
\cyrdje o1472
\CYRDZE o1472
\cyrdze o1472
\CYRDZHE o1472
\cyrdzhe o1472
\CYRE o1472
\cyre o1472
\CYREPS o1473
\cyreps o1472
\CYREREV o1473
\cyrerev o1473
\CYRERY o1473
\cyrery o1473
\CYRF o1473
\cyrf o1473
\CYRFITA o1474
\cyrfita o1473
\CYRG o1474
\cyrg o1474
\CYRGDSC o1474
\cyrgdsc o1474
\CYRGDSCHCRS o1474
\cyrgdschcrs o1474
\CYRGHCRS o1475
\cyrghcrs o1475
\CYRGHK o1475
\cyrghk o1475
\CYRGUP o1475
\cyrgup o1475
\CYRH o1475
\cyrh o1475
\CYRHDSC o1476
\cyrhdsc o1476
\CYRHHCRS o1476
\cyrhhcrs o1476
\CYRHHK o1476
\cyrhhk o1476
\CYRHRDSN o1477
\cyrhrdsn o1476
\CYRI o1477
\cyri o1477
\CYRIE o1477
\cyrie o1477
\CYRII o1477
\cyrii o1477
\CYRISHRT o1477
\cyrishrt o1477
\CYRISHRTDSC o1478
\cyrishrtdsc o1478
\CYRIZH o1478
\cyrizh o1478
\CYRJE o1478
\cyrje o1478
\CYRK o1478
\cyrk o1478
\CYRKBEAK o1479
\cyrkbeak o1479
\CYRKDSC o1479
\cyrkdsc o1479
\CYRKHCRS o1479
\cyrkhcrs o1479
\CYRKHK o1480
\cyrkhk o1479
\CYRKVCRS o1480
\cyrkvcrs o1480
\CYRL o1480
\cyrl o1480
\CYRLDSC o1480
\cyrldsc o1480
\CYRLHK o1481
\cyrlhk o1480
\CYRLJE o1481
\cyrlje o1481
\CYRM o1481
\cyrm o1481
\CYRMDSC o1481
\cyrmdsc o1481
\CYRMHK o1481
\cyrmhk o1481
\CYRN o1482
\cyrn o1482
\CYRNDSC o1482
\cyrndsc o1482
\CYRNG o1482
\cyrng o1482
\CYRNHK o1482
\cyrnhk o1482
\CYRNJE o1483
\cyrnje o1482
\CYRNLHK o1483
\cyrnlhk o1483
\CYRO o1483
\cyro o1483
\CYROTLD o1483
\cyrotld o1483
\CYRP o1483
\cyrp o1483
\CYRPHK o1484
\cyrphk o1484
\CYRQ o1484
\cyrq o1484
\CYRR o1484
\cyrr o1484
\CYRRDSC o1484

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\cyrrdsc o1484
\CYRRHK o1485
\cyrrhk o1484
\CYRRTICK o1485
\cyrrtick o1485
\CYRS o1485
\cyrs o1485
\CYRSACRS o1485
\cyrsacrs o1485
\CYRSCHWA o1486
\cyrschwa o1486
\CYRSDSC o1486
\cyrsdsc o1486
\CYRSEMISFTSN o1486
\cyrsemisftsn o1486
\CYRSFTSN o1487
\cyrsftsn o1487
\CYRSH o1487
\cyrsh o1487
\CYRSHCH o1487
\cyrshch o1487
\CYRSHHA o1487
\cyrshha o1487
\CYRT o1488
\cyrt o1488
\CYRTDSC o1488
\cyrtdsc o1488
\CYRTETSE o1488
\cyrtetse o1488
\CYRTSHE o1488
\cyrtshe o1488
\CYRU o1489
\cyru o1489
\CYRUSHRT o1489
\cyrushrt o1489
\CYRV o1489
\cyrv o1489
\CYRW o1489
\cyrw o1489
\CYRY o1489
\cyry o1489
\CYRYA o1490
\cyrya o1490
\CYRYAT o1490
\cyryat o1490
\CYRYHCRS o1490
\cyryhcrs o1490
\CYRYI o1490
\cyryi o1490
\CYRYO o1491
\cyryo o1490
\CYRYU o1491
\cyryu o1491
\CYRZ o1491
\cyrz o1491
\CYRZDSC o1491
\cyrzdsc o1491
\CYRZH o1491
\cyrzh o1491
\CYRZHDSC o1492
\cyrzhdsc o1492

D

\d o235, o395, o479, o759, o1215, o1423, o1424, o1425, o1426, o1429, o1430, o1431, o1432, o1433, o1434, o1435, o1436, o1437, o1438, o1439, o1440, o1441, o1442, o1443, o1444, o1445, o1446, o1447, o1448, o1449, o1450, o1451, o1452, o1453, o1454, o1455, o1456, o1457, o1458, o1459, o1460, o1461, o1462
\dag o312
\dagger o312, p131, p137, p145, p146, x375
\dashbox I309, I808, I825
\dashv x403
\date *588*
\date K9, K25
\day a185, P1011, P1142, P1231
\dblfigrule S767, S2325
\dblfloatpagefraction L287, L301, S2302
\dblfloatsep S753, S765, S1640, S1766, S2309
\dbltextfloatsep S222, S230, S769, S1639, S1765, S2309
\dbltopfraction L284, L298, S2301
\ddag o313
\ddagger o313, p132, p138, p145, p147, x374
\ddot x518
\ddots x513
\deadcycles n335, n368, D31, D100, D154, S299
debug commands:
\debug_resume: . g85, g345, g437, g752
\debug_suspend: g75, g337, g414, g747
\DebugHooksOff *109*, g942, g982
\DebugHooksOn *109*, g942, g981
\DebugShipoutsOff *703*, R328, R359
\DebugShipoutsOn *703*, R328, R358
\DeclareCommandCopy . . *86*, f460, f461, f463
\DeclareCurrentRelease A724, A727, P1466
\DeclareDefaultHookLabel *104*, *104*, *151*, *151*, g907
\DeclareDefaultHookRule *106*, g946, g985
\DeclareEmphSequence *382*
\DeclareEmphSequence *383*, w391, w427, w428, w440

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\DeclareEncodingSubset A60, A61, A324, A325, A326, A327, A328, A329, A330, A331, A332, A333, A334, A335, A336, A337, A338, A339, A340, A341, A342, A343, A344, A345, A346, A347, A348, A349, A350, A351, A352, A353, A354, A355, A356, A357, A358, A359, A360, A361, A362, A363, A364, A365, A366, A367, A368, A369, A370, A371, A372, A373, A374, A375, A376, A377, A378, A379, A380, A381, A382, A383, A384, A385, A386, A387, A388, A389, A390, A391, A392, A393, A394, A395, A396, A397, A398, A399, A400, A401, A402, A403, A404, A405, A406, A407, A408, A409, A410, A411, A412, A413, A414, A415, A416, A417, A418, A419, A420, A421, A422, A423, A424, A425, A426, A427, A428, A429, A430, A431, A432, A433, A434, A435, A436, A437, A438, A439, A440, A441, A442, A443, A444, A445, A446, A447, A448, A449, A450, A451, A452, A453, A454, A455, A456, A457, A458, A459, A460, A461, A462, A463, A464, A465, A466, A467, A468, A469, A470, A471, A472, A473, A474, A475, A476, A477, A478, A479, A480, A481, A482, A483, A484, A485, A486, A487, A488, A489, A490, A491, A492, A493, A494, A495, A496, A497, A498, A499, A500, A501, A534, A536, A732, A733, A734, A735, A774, A781, A782, A783, A784, A1060, A1061, A1062, A1063, A1064, A1065, A1066, A1067, A1068, A1069, A1070, A1071, A1072, A1073, A1074, A1075, A1076, A1077, A1078, A1079, A1080, A1081, A1082, A1083, A1084, A1085, A1086, A1087, A1088, A1089, A1090, A1091, A1092, A1093, A1094, A1095, A1096, A1097, A1098, A1099, A1100, A1101, A1102, A1103, A1104, A1105, A1106, A1107, A1108, A1109, A1110, A1111, A1112, A1113, A1114, A1115, A1116, A1117, A1118, A1119, A1120

\DeclareErrorFont . r448, v264, w548, x49

\DeclareFixedFont r75

\DeclareFontEncoding o377, o460, o710, o732, o738, o824, o1022, r118, x126, x127, x128, x129

\DeclareFontEncodingDefaults r168, u90, u91, x36

\DeclareFontFamily *283*, r93, u85, u86

\DeclareFontFamilySubstitution . . . r408

\DeclareFontSeriesChangeRule *294*, *294*, s5, s6, s8, s9, s10, s11, s12, s13, s14, s15, s16, s17, s18, s19, s20, s21, s22, s23, s24, s25, s26, s27, s28, s29, s30, s31, s32, s33, s34, s35, s36, s37, s38, s39, s40, s41, s42, s43, s44, s45, s46, s47, s48, s49, s50, s51, s52, s53, s54, s55, s56, s57, s58, s59, s60, s61, s62, s63, s64, s65, s66, s67, s68, s69, s70, s71, s72, s73, s74, s75, s76, s77, s78, s79, s80, s81, s82, s83, s84, s85, s86, s87, s88, s89, s90, s91, s92, s93, s94, s95, s96, s97, s98, s99, s100, s101, s102, s103, s104, s105, s106, s107, s108, s109, s110, s111, s112, s113, s114, s115, s116, s117, s118, s119, s120, s121, s122, s123, s124, s125, s126, s127, s128, s129, s130, s131, s132, s133, s134, s135, s136, s137, s138, s139, s140, s141, s142, s143, s144, s145, s146, s147, s148, s149, s150, s151, s152, s153, s154, s155, s156, s157, s158, s159, s160, s161, s162, s163, s164, s165, s166, s167, s168, s169, s170, s171, s172, s173, s174, s175, s176, s177, s178, s179, s180, s181, s182, s183, s184, s185, s186, s187, s188, s189, s190, s191, s192, s193, s194, s195, s196, s197, s198, s199, s200, s201, s202, s203, s204, s205, s206, s207, s208, s209, s210, s211, s212, s213, s214, s215, s216, s217, s218, s219, s220, s221, s222, s223, s224, s225, s226, s227, s228, s229, s230, s231, s232, s233, s234, s235, s236, s237, s238, s239, s240, s241, s242, s243, s244, s245, s246, s247, s248, s249, s250, s251, s252, s253, s254, s255, s256, s257, s258, s259, s260, s261, s262, s263, s264, s265, s266, s267, s268, s269, s270, s271, s272, s273, s274, s275, s276, s277, s278, s279, s280, s281, s282, s283, s284, s285, s286, s287, s288, s289,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

s290, s291, s292, s293, s294, s295, s296, s297, s298, s299, s300, s301, s302, s303, s304, s305, s306, s307, s308, s309, s310, s311, s312, s313, s314, s315, s316, s317, s318, s319, s320, s321, s322, s323, s324, s325, s326, s327, s328, s329, s330, s331, s332, s333, s334, s335, s336, s337, s338, s339, s340, s341, s342, s343, s344, s345, s346, s347, s348, s349, s350, s351, s352, s353, s354, s355, s356, s357, s358, s359, s360, s361, s362, s363, s364, s365, s366, s367, s368, s369, s370, s371, s372, s373, s374, s375, s376, s377, s378, s379, s380, s381, s382, s383, s384, s579, s586
`\DeclareFontSeriesDefault` *368*
`\DeclareFontSeriesDefault` *368*, *369*, *373*, *374*, *374*, *380*, w34, w35, w164, w166
`\DeclareFontShape` r18, r417, r418, r419, r420, r421, r422, r423, r424, r425, r426, r427, r428, r429, r430, r431, r432, r433, r434, r435, r436, r437, u25, u27, u81, u82
`\DeclareFontShapeChangeRule` s447, s464, s465, s466, s467, s468, s469, s470, s471, s472, s473, s474, s475, s476, s477, s478, s479, s480, s481, s482, s483, s484, s485, s486, s487, s488, s489, s490, s491, s492, s493, s494, s495, s496, s497, s498, s499, s500, s501, s502, s503, s504, s505, s506, s507, s508, s509, s510, s511, s512, s513, s514, s515, s516, s517, s518, s519, s520, s594
`\DeclareFontSubstitution` o739, o825, r141, x26, x34, x37, x38, x130, x131, x132, x133
`\DeclareHookRule` *100*, *105*, *106*, *108*, *110*, g944, g984, D39
`\DeclareHookrule` *102*, *105*
`\DeclareMathAccent` v605, x516, x517, x518, x519, x520, x521, x522, x523, x524, x525, x526, x527, x528
`\DeclareMathAlphabet` u119, u123, u125, u132, v431, v594, x151, x152, x153, x154
`\DeclareMathAlphabetCharacter` . . . v780
`\DeclareMathDelimiter` *407*, v782, x255, x256, x257, x258, x259, x260, x263, x265, x266, x563, x565, x567, x569, x571, x574, x576, x578, x580, x582, x584, x586, x588, x590, x592, x594, x596, x598, x600, x602, x604, x606, x608, x610, x612
`\DeclareMathRadical` v917, x529
`\DeclareMathSizes` . . r205, r211, r233, x157, x158, x159, x160, x161, x162, x163, x164, x165, x166, x167, x168
`\DeclareMathSizes*` r205
`\DeclareMathSymbol` *403*, v718, v781, v798, x169, x170, x171, x172, x173, x174, x175, x176, x177, x178, x179, x180, x181, x182, x183, x184, x185, x186, x187, x188, x189, x190, x191, x192, x193, x194, x195, x196, x197, x198, x199, x200, x201, x202, x203, x204, x205, x206, x207, x208, x209, x210, x211, x212, x213, x214, x215, x216, x217, x218, x219, x220, x221, x222, x223, x224, x225, x226, x227, x228, x229, x230, x231, x232, x233, x234, x235, x236, x237, x238, x239, x240, x241, x242, x243, x244, x245, x246, x247, x248, x249, x250, x251, x261, x262, x264, x268, x269, x270, x271, x272, x273, x274, x275, x276, x277, x278, x279, x280, x281, x282, x283, x284, x285, x286, x287, x288, x289, x290, x291, x292, x293, x294, x295, x296, x297, x298, x299, x300, x301, x302, x303, x304, x305, x306, x307, x308, x309, x310, x311, x312, x313, x314, x315, x316, x317, x318, x319, x320, x321, x322, x323, x324, x325, x327, x328, x329, x330, x331, x332, x333, x334, x341, x342, x343, x344, x345, x346, x347, x349, x350, x351, x352, x353, x354, x356, x357, x358, x359, x360, x361, x364, x365, x366, x367, x370, x371, x372, x373, x374, x375, x376, x377, x378, x379, x380, x381, x382, x383, x384, x385, x386, x387, x388, x389, x390, x391, x392, x393, x394, x395, x396, x397, x398, x399, x400, x401, x402, x403, x404, x405, x406, x407, x408, x409, x410, x411, x414, x415, x418, x419, x420, x421, x422, x423, x424, x425, x426, x427, x428, x429, x430, x432, x433, x434, x435, x436, x437, x438, x441, x442, x443, x445, x446, x447, x448, x449, x450, x451, x452, x453, x454, x455, x477, x479, x501, x502, x503, x553, x554, x555, x556, x614, x615, x616
`\DeclareMathVersion` v277, w2, w3

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

\DeclareOldFontCommand z125, z141
\DeclareOption *636*
\DeclareOption o1495, t29, t37, t45, t53, t56, t60, A732, A733, A734, A735, A736, A737, A738, A739, A740, A781, A782, A783, A784, A785, A787, A788, P378, P908
\DeclareOption* *636*
\DeclareOption* P378
\DeclarePreloadSizes r185, u95, u96, y19, y21, y22, y23, y25, y26, y27, y28, y29, y30, y34, y38, y43, y45, y49, y50, y53, y54, y57, y58, y64
\DeclareRelease A723, A726, P1373
DeclareRelease commands:
\DeclareRelease: P1376
\DeclareRobustCommand ... *82*, *83*, *86*, *90*, *90*, *90*, *91*, *92*, *93*, *228*, *309*, *469*, f221, f728, f729, f735, f750, j4, j11, j30, j57, l7, l8, l9, l10, l11, l68, l92, l330, l406, l420, l437, l461, m2, m3, m13, n407, n477, o150, o158, o307, o310, o311, o312, o313, o314, o316, o318, o320, p155, q19, q20, q21, r251, r279, r280, r281, r282, r287, r289, r307, r612, r621, s386, s387, s449, s454, s459, s521, s522, s581, s582, t113, w4, w7, w10, w13, w16, w19, w22, w25, w28, w198, w218, w237, w242, w247, w279, w290, w301, w306, w311, w324, w327, w330, w333, w336, w350, w399, w400, w435, w441, w463, w465, w474, w476, w483, w499, w507, w521, x335, x336, x337, x348, x355, x412, x413, x444, x456, x460, x463, x468, x470, x472, x475, x478, x480, x481, x483, x485, x487, x489, x491, x493, x495, x497, x499, x505, x507, x509, x512, x530, x533, x536, x540, x544, x547, x550, x557, x560, x617, x618, x619, x624, x626, x628, x630, z3, z126, A4, A11, A45, A1045, C47, C59, D161, D217, D343, D348, D353, D363, D367, D371, D452, D456, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E22, E23, E24, E25, E26, E27, E28, E29, E30, E31, E32, E33, E34, E35, E39, E41, E42, E43, E44, E45, E46, E47, E48, E49, E50, E51, E52, E81, E82, E83, E84, E126, E154, E156, E160, E205, E207, E209, E211, E214, E215, E217, E224, E226, E227, E238, E261, E263, E279, E290, E340, E341, E342, E415, E432, E456, G7, G24, G101, G114, G137, G138, G154, G165, G219, G367, G385, G393, G429, G430, G431, G432, G433, G434, H139, H142, H154, K7, K8, K9, K10, K14, K207, L378, L399, N12, O22, O30, O53, O55, O57, O64, P864, P865, P866, P872, P873, P1496, Q145, Q177, R361, S87, U463, U473, I124, I127, I130
\DeclareSizeFunction t371, t444, t445, t456, t457, t461, t462, t468, t469, t497, t511, t512, t519, t520
\DeclareSymbolFont u136, v312, x141, x142, x143, x144
\DeclareSymbolFontAlphabet v991, x148, x149, x150
\DeclareTextAccent o64, o378, o379, o380, o381, o382, o383, o384, o385, o386, o387, o388, o461, o462, o463, o464, o465, o466, o467, o468, o469, o470, o471, o735, o740, o741, o742, o743, o744, o745, o746, o747, o748, o749, o750, o832, o833, o834, o835, o836, o837, o838, o839, o840, o841, o842, o843, o844, o845, o846
\DeclareTextAccentDefault o185, o226, o227, o228, o229, o230, o231, o232, o233, o234, o235, o236, o237, o238, o239, o279, o282, A597, A598, A849, A850, A851, A852, A853, A854, A855, A856, A857, A858, A859, A860, A861
\DeclareTextCommand o3, o58, o65, o83, o389, o392, o395, o411, o412, o419, o421, o423, o425, o431, o433, o435, o442, o472, o475, o479, o482, o484, o487, o489, o491, o507, o724, o751, o753, o756, o759, o792, o799, o826, o829, o876, o1027, o1053, o1055, o1057, o1181, o1218, A303, A304, A305, A306, A307, A308, A309, A310, A311, A312, A541, A548, A555, A669
\DeclareTextCommandDefault o57, o186, o188, o283, o286, o287, o288, o290, o292, o296, o300, o301, o303, o304, o305, o306, o326, o355, A175, A177, A180, A181, A182, A183, A184, A185, A186, A187, A188, A189, A190, A191, A192, A193, A194, A195, A197, A198, A199,

A200, A201, A202, A203, A204, A205, A206, A207, A208, A209, A210, A211, A212, A214, A215, A216, A217, A218, A219, A220, A221, A222, A223, A224, A225, A226, A227, A228, A229, A230, A231, A232, A233, A234, A235, A236, A237, A238, A239, A240, A241, A242, A243, A244, A245, A246, A247, A248, A249, A250, A251, A252, A253, A254, A255, A256, A257, A258, A259, A260, A261, A262, A264, A265, A266, A267, A268, A269, A270, A271, A272, A273, A274, A275, A276, A277, A278, A279, A280, A281, A282, A283, A284, A285, A287, A288, A289, A290, A291, A292, A293, A570, A577, A578, A583, A589, A915, A917, A918, A920, A922, A924, A926, A928, A930, A932, A934, A936, A938, A940, A942, A944, A946, A948, A950, A952, A954, A956, A958, A960, A962, A964, A966, A968, A970, A972, A974, A976, A978, A980, A982, A984, A986, A988, A990, A992, A994, A996, A998, A1000, A1002, A1004, A1006, A1008, A1010, A1012, A1014, A1016, A1018, A1020, A1022, A1024, A1026, A1028, A1030, A1032, A1035

`\DeclareTextComposite` o76, o449, o450, o581, o582, o583, o584, o585, o586, o587, o588, o589, o590, o591, o592, o593, o594, o595, o596, o597, o598, o599, o600, o601, o602, o603, o604, o605, o606, o607, o608, o609, o610, o611, o612, o613, o614, o615, o616, o617, o618, o619, o620, o621, o622, o623, o624, o625, o626, o627, o628, o629, o630, o631, o632, o633, o634, o635, o636, o637, o638, o639, o640, o641, o642, o643, o644, o645, o646, o647, o648, o649, o650, o651, o652, o653, o654, o655, o656, o657, o658, o659, o660, o661, o662, o663, o664, o665, o666, o667, o668, o669, o670, o671, o672, o673, o674, o675, o676, o677, o678, o679, o680, o681, o682, o683, o684, o685, o686, o687, o688, o689, o690, o691, o806, o807, o808, o809, o810, o811, o812, o813, o814, o815, o816, o817, o818, o819, o820, o821

`\DeclareTextCompositeCommand` o76, o428, o451, o452, o453, o454, o456, o692, o693, o695, o698, o699, o700, o701, o702, o703, o704, o705, o706, o789, o1033

`\DeclareTextFontCommand` z1, z15, z16, z17, z18, z19, z20, z21, z22, z23, z24, z29, z30, z31, z42, z140

`\DeclareTextSymbol` o3, o398, o399, o400, o401, o402, o403, o404, o405, o406, o407, o408, o409, o410, o413, o414, o415, o416, o417, o418, o523, o524, o525, o526, o527, o528, o529, o530, o531, o532, o533, o534, o535, o536, o537, o538, o540, o541, o542, o543, o544, o545, o546, o547, o548, o549, o550, o551, o552, o553, o554, o555, o556, o557, o558, o559, o560, o561, o562, o563, o564, o565, o566, o567, o568, o569, o570, o571, o572, o573, o574, o575, o576, o577, o578, o579, o580, o711, o712, o713, o714, o715, o716, o717, o718, o719, o720, o721, o722, o723, o733, o734, o762, o763, o764, o765, o766, o767, o768, o770, o771, o772, o773, o774, o775, o776, o777, o778, o779, o780, o781, o782, o783, o784, o785, o786, o787, o788, o847, o848, o849, o850, o851, o852, o853, o854, o855, o856, o857, o858, o859, o860, o861, o862, o863, o864, o865, o866, o867, o868, o869, o870, o871, o872, o873, o874, o875, o882, o883, o884, o885, o886, o887, o888, o889, o890, o891, o892, o893, o894, o895, o896, o897, o898, o899, o900, o901, o902, o903, o904, o905, o906, o907, o908, o909, o910, o911, o912, o913, o914, o915, o916, o917, o918, o919, o920, o921, o922, o923, o924, o925, o926, o927, o928, o929, o930, o931, o932, o933, o934, o935, o936, o937, o938, o939, o940, o941, o942, o943, o944, o945, o946, o947, o948, o949, o950, o951, o952, o953, o954, o955, o956, o957, o958, o959, o960, o961, o1059, o1060, o1061, o1062, o1063, o1064, o1065, o1066, o1067, o1068, o1069, o1070, o1071, o1072, o1073, o1074, o1075, o1076, o1077, o1078, o1079, o1081, o1082, o1083, o1084, o1085, o1086, o1087, o1088, o1089,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

o1090, o1091, o1092, o1093, o1094, o1095, o1097, o1098, o1099, o1100, o1101, o1102, o1103, o1104, o1105, o1106, o1107, o1108, o1109, o1110, o1111, o1112, o1113, o1114, o1115, o1116, o1117, o1118, o1119, o1120, o1121, o1122, o1123, o1124, o1125, o1126, o1127, o1128, o1129, o1130, o1131, o1132, o1133, o1134, o1135, o1136, o1137, o1138, o1139, o1140, o1141, o1142, o1143, o1144, o1145, o1146, o1147, o1148, o1149, o1150, o1151, o1152, o1153, o1154, o1155, o1156, o1157, o1158, o1159, o1160, o1161, o1162, o1163, o1164, o1165, o1166, o1167, o1168, o1169, o1170, o1171, o1172, o1173, o1174, o1175, o1176, o1177, o1179, o1180, o1192, o1193, o1194, o1195, o1196, o1197, o1198, o1199, o1200, o1201, o1202, A295, A296, A297, A298, A299, A300, A301, A302, A313, A314, A315, A316, A317, A318, A319, A320, A321, A322, A515, A516, A517, A518, A519, A520, A521, A522

`\DeclareTextSymbolDefault` o185, o240, o241, o242, o243, o244, o245, o246, o247, o248, o249, o250, o251, o252, o253, o254, o255, o256, o257, o258, o259, o260, o261, o262, o263, o264, o265, o266, o267, o268, o269, o270, o271, o272, o273, o274, o275, o276, o277, o278, o280, o281, o291, A134, A136, A138, A140, A141, A142, A143, A144, A145, A146, A147, A148, A149, A150, A151, A152, A153, A154, A155, A156, A157, A158, A159, A160, A161, A162, A163, A164, A165, A166, A167, A168, A169, A170, A171, A172, A173, A174, A503, A504, A505, A506, A507, A508, A509, A510, A523, A524, A525, A526, A527, A528, A529, A530, A539, A540, A558, A559, A560, A561, A562, A563, A564, A566, A862, A863, A864, A865, A866, A867, A868, A869, A870, A871, A872, A873, A874, A875, A876, A877, A878, A879, A880, A881, A882, A883, A884, A885, A886, A887, A888, A889, A890, A891, A892, A893, A894, A895, A896, A897, A898, A899, A900, A901, A902, A903, A904, A905, A906, A907, A908, A909, A910, A911, A912, A913, A914

`\DeclareUnicodeAccent` o1026, o1203, o1204, o1205, o1206, o1207, o1208, o1209, o1210, o1211, o1212, o1213, o1214, o1215, o1216, o1217

`\DeclareUnicodeCharacter` U282

`\DeclareUnicodeComposite` o1031, o1222, o1223, o1224, o1225, o1226, o1227, o1228, o1229, o1230, o1231, o1232, o1233, o1234, o1235, o1236, o1237, o1238, o1239, o1240, o1241, o1242, o1243, o1244, o1245, o1246, o1247, o1248, o1249, o1250, o1251, o1252, o1253, o1254, o1255, o1256, o1257, o1258, o1259, o1260, o1261, o1262, o1263, o1264, o1265, o1266, o1267, o1268, o1269, o1270, o1271, o1272, o1273, o1274, o1275, o1276, o1277, o1278, o1279, o1280, o1281, o1282, o1283, o1284, o1285, o1286, o1287, o1288, o1289, o1290, o1291, o1292, o1293, o1294, o1295, o1296, o1297, o1298, o1299, o1300, o1301, o1302, o1303, o1304, o1305, o1306, o1307, o1308, o1309, o1310, o1311, o1312, o1313, o1314, o1315, o1316, o1317, o1318, o1319, o1320, o1321, o1322, o1323, o1324, o1325, o1326, o1327, o1328, o1329, o1330, o1331, o1332, o1333, o1334, o1335, o1336, o1337, o1338, o1339, o1340, o1341, o1342, o1343, o1344, o1345, o1346, o1347, o1348, o1349, o1350, o1351, o1352, o1353, o1354, o1355, o1356, o1357, o1358, o1359, o1360, o1361, o1362, o1363, o1364, o1365, o1366, o1367, o1368, o1369, o1370, o1371, o1372, o1373, o1374, o1375, o1376, o1377, o1378, o1379, o1380, o1381, o1382, o1383, o1384, o1385, o1386, o1387, o1388, o1389, o1390, o1391, o1392, o1393, o1394, o1395, o1396, o1397, o1398, o1399, o1400, o1401, o1402, o1403, o1404, o1405, o1406, o1407, o1408, o1409, o1410, o1411, o1412, o1413, o1414, o1415, o1416, o1417, o1418, o1419, o1420, o1421, o1422, o1423, o1424, o1425, o1426, o1427, o1428, o1429, o1430, o1431, o1432, o1433, o1434, o1435, o1436, o1437, o1438, o1439, o1440, o1441, o1442, o1443, o1444, o1445, o1446, o1447, o1448, o1449, o1450, o1451, o1452,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

o1453, o1454, o1455, o1456, o1457, o1458, o1459, o1460, o1461, o1462
\def *153*, *269*, *369*, *375*, *395*, *505*
\defaulthyphenchar b333, f738, f753
\defaultscriptratio r595, r602
\defaultscriptscriptratio . . . r596, r602
\defaultskewchar b334
\deg E34
\delcode v915
\delimiter v828, v898, v909
\delimiterfactor b335
\delimitershortfall b345
\Delta x298
\delta x271
\depth G32, G35
\det E30
\detokenize *96*, *202*, *688*, f527, f545, f609, n255, o981, o1024, s396, w413, A65, D119, P659, P981, P982, P986, P1112, P1113, P1117
\DH o525, o1103, U483
\dh o535, o1109, U483
\Diamond w540
\diamond x380
\diamondsuit x332
\dim E28
dim commands:
  \dim_new:N . . . R123, R124, R125, R126
  \dim_set:Nn . . R117, R118, R119, R120
  \dim_use:N R431, R432, R433
  \c_max_dim . . . R137, R163, R188, R215
  \c_zero_dim R143, R144, R145, R151, R169, R170, R171, R194, R195, R196, R223, R224, R225, R253, R255, R256
\dimendef . . b42, b43, b44, b52, b91, d218
\dimenzero d218
\dimexpr . . . o500, o516, o1185, o1188, I13
\directlua a9, a12, a17, a20, a25, b65, b81, b105, b256, d2, d14, d29, d205, d223, d248, d253, d257, f34, f649, f734, o994, P965, P1096
disable commands:
  disable_callback d807
\disable_callback *42*
\DiscardShipoutBox *701*, *702*, *704*, R326, R357, R421
\discretionary f736, f751, f761, E238
\displaylines E190
displaymath (environment) E332
\displaymath E334
\displaystyle x532, x535, x539, x541, E62, E197, E358, E361, E414, E439, E451, E479, E503, E506
\displaywidowpenalty b325
\displaywidth . . . E197, E357, E426, E482
\div x383
\DJ o526, o1113, U483
\dj o536, o1114, U483
\do a74, a75, a126, b13, b14, f69, i3, i7, i16, i26, n68, n71, n136, n139, n191, n194, n234, n305, n352, n427, n465, n608, n614, z90, D417, D438, D546, D556, D562, D568, G57, G76, H244, H269, L65, L134, N16, N41, P216, P233, P405, P419, P435, P440, P454, P459, P480, P490, P897, P934, P1014, P1065, P1145, P1234, P1511, I104, I116, I123, I203, I337, I339, I362, I365, I394, I396, I417, I422, I478, I506, I535, I731, I787
\DocInput t8, x5, y5, T4
\document *193*
document (environment) D8
\document *116*, *117*, *310*, *374*, *467*, n11, D172, N40
\documentclass *104*, *634*, d16, t2, x2, y2, P497, P504, P547, P550, P679, P806, P917, T2
\documentstyle P502, P917
\dospecials a74, a126, b13, D417, D438, D546, D556, P1014, P1145, P1234
\dot x525
\doteq x469
\dotfill b446, f776, f797
\dots o320, o322
\doublehyphendemerits b328
\doublerulesep H311, H338, H362
\Downarrow x586
\downarrow x580
\downbracefill x538, x557
\dump U548

E

\E P1021, P1024, P1051, P1152, P1155, P1182, P1241, P1244, P1272
\edef *305*, *305*, *347*, *374*
\egroup b383
\eject b413
\ell x311
else commands:
  \else: g376, g391, g530, g781, g857, Q232
\em *383*, *384*, w400, w427, z42
\emergencystretch O59, O65
\emforce *382*
\emforce *382*, *383*, *384*, *384*, *384*, w396, w423, w432

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\eminnershape *382*, *383*, w404, w410, w427
\emph *383*, z42
\empty b381
\emptyset x318
\emreset *382*
\emreset *382*, w396, w399, w430
\emrest w399
\encodingdefault *373*, d252, d267, d275, o973, o1496, o1529, o1530, v269, w508, w522, x50
\end *100*, *116*, *117*, *118*, *130*, *153*, *468*, *469*, *470*, *506*, a69, f23, f671, j200, t9, x6, y6, D179, D186, D188, D216, D242, D396, D397, E460, E469, F112, K15, K17, P1029, P1033, P1039, P1160, P1164, P1170, P1249, P1253, P1259, R311, T5
\end⟨*env*⟩ *116*
\endarray H171
\endcenter D338
\endcsname *103*, *691*, *692*
\enddisplaymath E335
\enddocument *114*, *117*, *462*, *699*, *699*, *715*, *715*, *716*, D8, D67, D69, R276
\endenumerate F240
\endeqnarray E366, E413
\endequation E338
\endfilecontents P921
\endflushleft D388
\endflushright D390
\endgraf b378
\endgroup *195*, *467*, *693*
\EndIncludeInRelease a22, a50, a306, a316, b87, b101, b118, b123, b133, b137, b147, b150, b167, b181, b185, b219, b224, b232, b240, b288, b300, b491, b498, b545, b550, c97, c103, d226, d249, d272, d276, e8, e99, e120, e128, e140, e149, f12, f18, f60, f64, f312, f340, f368, f373, f392, f406, f447, f457, f487, f496, f510, f515, f613, f628, f660, f669, f719, f726, f748, f759, f762, f790, f811, g962, g990, l20, l30, l64, l77, l85, l90, l103, l109, l151, l165, l177, l188, l205, l217, l251, l268, l306, l328, l365, l398, l431, l435, l444, l450, l464, l471, n84, n141, n196, n257, n278, n290, n339, n373, n437, n475, n488, n492, n510, n521, n531, n570, n597, o103, o123, o168, o175, o330, o352, o367, o375, p28, p33, p89, p100, p142, p148, p168, p171, q10, q14, r52, r73, r231, r248, r401, r406, r441, r446, r463, r481, r520, r553, r669, r681, s577, s607, s616, s620, s623, t503, t509, u21, u143, v77, v105, v168, v198, v230, v654, v696, v709, v716, v904, v912, w162, w184, w266, w318, w344, w380, w389, w426, w438, w445, w479, w485, w517, w531, x80, x90, x109, x118, x633, x640, z33, z40, A41, A55, A532, A719, C50, C61, C72, D65, D106, D139, D152, D214, D264, D280, D294, D303, D313, D320, D330, D335, D359, D375, D384, D422, D443, D474, D484, D502, D517, D529, D551, D559, E86, E95, E117, E124, E143, E147, E162, E170, E219, E236, E266, E274, E303, E330, E393, E402, E442, E454, E463, E472, F132, F137, G13, G22, G69, G86, G105, G112, G169, G176, G224, G232, G283, G301, G369, G374, G397, G404, H64, H69, H156, H164, H226, H231, K19, K29, K167, K173, K178, K209, K231, L104, L172, L231, L246, L293, L306, L387, L393, L402, L406, L415, L421, L425, O39, O46, P20, P25, P51, P63, P73, P82, P88, P93, P115, P122, P154, P160, P179, P190, P228, P245, P263, P275, P296, P309, P322, P332, P362, P371, P449, P464, P485, P494, P536, P543, P621, P649, P781, P838, P868, P875, P1054, P1184, P1274, Q14, Q23, Q87, Q137, Q172, Q184, Q192, Q236, Q247, Q254, Q280, Q293, Q297, Q320, Q340, Q365, Q410, Q426, Q433, Q463, R343, R387, R404, R408, S53, S62, S178, S196, S365, S370, S418, S464, S650, S708, S811, S830, S893, S911, S953, S974, S1216, S1385, S1467, S1561, S1683, S1810, S1929, S1957, S2177, S2195, S2242, S2286, U15, U19, U37, U55, U65, U72, U80, U131, U155, U225, U229, U338, U385, I15, I19, I38, I47, I68, I76, I88, I96, I111, I121, I150, I155, I171, I182, I249, I263, I368, I425, I463, I469, I500, I530, I555, I571, I582, I595, I603, I624, I641, I651, I655, I745, I800, I819, I836
\EndIncludeRelease c73

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\enditemize F251
\endline b378, E175
\endlinechar *464*, a92, a93, a94, a204, f37, f39, f44, n545, n579, P337, P338, P339
\endlist F98, F240, F251
\endlrbox G136
\endmath E333
\endminipage G335
\endpicture I49
\endsloppypar O63
\endtabbing H84
\endtabular H171
\endtabular* H171
\endtrivlist D338, D388, D390, D446, E484, F100, F101, H85, J39
\endverbatim D445, D472
\enlargethispage S1855
\enlargethispage* S1855
\enskip l473
\enspace l461, l469
\ensuremath p144, E415, L385, L392, L413, L420
enumerate (environment) F231
\enumerate F231
environments:
  array H168
  center D337
  displaymath E332
  document D8
  enumerate F231
  eqnarray E344, E485
  eqnarray* E411
  equation E336, E473
  filecontents *634*, P921
  flushleft D387
  flushright D389
  itemize F242
  list F34
  lrbox *512*
  math E332
  minipage *513*
  picture I21
  sloppypar O62
  tabbing H71
  tabular H174
  thebibliography *625*
  trivlist F89
  verbatim D445
  verbatim* D469
\epsilon x272
eqnarray (environment) E344, E485
\eqnarray E349, E412
eqnarray* (environment) E411
\eqno E338
equation (environment) E336, E473
\equation E337
\equiv x448
\errhelp a217, c31, j39, j66, T12, U239, U538
\errmessage a4, a58, a222, b164, b178, b304, c32, d63, e60, e75, j47, j72, r492, r527, t379, t479, u65, T16, U49, U241
\ERROR g516, g517
\errorcontextlines b338, b465, b481, b496, b509, b526, j163
\errorstopmode *108*, *111*, b454, U547
\escapchar *205*
\escapechar *91*, d204, f126, f169, f173, f181, f275, f276, f300, f437, f525, f543, n392, r338, r562, t183, v58, v86, v147, v178, v209, v253, Q264, Q285
\eta x274
\etatcatcode d859
\eTeXversion a57
\evensidemargin S73, S613, S672
\everycr . . . b434, E192, E195, E357, E500
\everydisplay r316, r317, r322
\everyjob . . c37, c42, c47, d208, d254, d255, v273, U314, U508, U509, U511
\everymath r315, r317, r320
\everypar *171*
\everypar *506*, n45, n114, n172, r615, r628, r675, D155, D419, D441, F129, F131, F135, F136, F180, F197, G273, G294, H81, K48, K96, K107, K127, K136, L187, S165, S192, S1151, S1317
\EveryShipout *704*, R443
\ExecuteOptions t57, t70, A752, A789, P473
\exhyphenpenalty b320, b407
\exists x324
\exp E31
exp commands:
  \exp_after:wN g134, g135, g529, g913, Q72, Q227, Q228
  \exp_args:Nc g756
  \exp_args:Ne g223, g224, g804, g805, Q46, Q96
  \exp_args:Nee Q126
  \exp_args:Nf Q32, Q34, Q330
  \exp_args:Nnnv g80
  \exp_args:NNx g319, g433
  \exp_args:No g699, g741
  \exp_args:NV Q53
  \exp_args:Nv g650, g723
  \exp_last_unbraced:NNNNo g137

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\exp_not:N g145, R56, R87
\exp_not:n g406, Q203, Q216
\exp_stop_f: g373, g381, g855
\expandafter *469, 470, 691*
expandafter commands:
\expandafter: z88, H269
\expanded *125*, P659
\ExplSyntaxOff e139, g960, Q12, Q85, Q135, Q234, Q245, Q257, Q338, Q363, Q408, Q461, Q511, R341, R395, R434
\ExplSyntaxOn e131, g5, Q7, Q29, Q91, Q199, Q242, Q326, Q346, Q371, Q440, Q504, R5, R393, R430
\extracolsep H167
\extrafloats b152, b189, b274

F

\fam b98, d22, d26, d38, r15
\familydefault *373*, o1530, v270, w276, w509, w523, x120
\fbox *512*
\fbox G154, G167, G174
\fboxrule G152, G188, G191, G197, G199, G206, G207, U84
\fboxsep G152, G158, G187, G192, G202, G204, U83
fi commands:
\fi: g377, g385, g391, g532, g725, g783, g859, Q232
\filbreak b411
file commands:
\file_full_name:n *688*, Q35
\file_parse_full_name_apply:nN Q32, Q47, Q89, Q92, Q94
\l_file_search_path_seq *680, 688, 693*
file internal commands:
__file_parse_full_name_area:nw Q101, Q104, Q108
__file_parse_full_name_auxi:nN Q96, Q99
__file_parse_full_name_base:nw Q107, Q110, Q122
__file_parse_full_name_tidy:nnnN Q117, Q118, Q120, Q124
file/after *679*
file/after/... ..... *679*
file/before *679*
file/before/... ..... *679*
filecontents (environment) *634*, P921
\filecontents P921
filehook internal commands:
__filehook_clear_replacement_- flag: Q353, Q452
__filehook_drop_extension:N Q44, Q454
__filehook_drop_extension_- aux:nnn Q54, Q57
__filehook_file_name_compose:nnn Q210, Q342, Q361
__filehook_file_parse_full_- name:nN ... *685, 685, 694*, Q28, Q30, Q53, Q229, Q330, Q348, Q350
__filehook_file_pop: ... Q59, Q458
__filehook_file_pop_assign:nnnn Q59, Q460
__filehook_file_push: ... Q59, Q456
__filehook_file_subst_begin:nnn *694*, Q348, Q358
__filehook_file_subst_cycle_- error:NN *695*, Q393, Q398
__filehook_file_subst_loop:NN *695*, Q367
__filehook_file_subst_tortoise_- hare:nn *694, 695*, Q360, Q367
__filehook_full_name:nn Q30
__filehook_if_file_replaced:TF *694*, Q353, Q450
__filehook_if_no_extension:nTF Q44, Q442
\g__filehook_input_file_seq *685*, Q59
\l__filehook_internal_tl Q59
__filehook_log_file_record:n . Q474
\g__filehook_nesting_level_int Q471, Q476, Q479, Q480, Q486
__filehook_normalize_file_- name:w Q342, Q448
__filehook_resolve_file_subst:w Q342, Q446
__filehook_set_curr_file:nNN Q328, Q444
__filehook_set_curr_file_- assign:nnnNN Q30, Q331, Q333
__filehook_subst_add:nn *689*, Q198, Q200, Q243
__filehook_subst_empty_name_- chk:NN Q200
__filehook_subst_file_normalize:Nn Q200
__filehook_subst_remove:n Q200, Q244
\filename@parse *1, 6*
\filesize e53, e107
\fill l454
\finalhyphendemerits b329, D346, D350, D356
\firstmark O51, S647, S706, S2217

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

flag internal commands:
\flag␣__filehook_file_replaced . *694*
flag␣__filehook_file_replaced . Q353
flag commands:
\flag_clear:n Q357
\flag_if_raised:nTF Q355, Q376
\flag_new:n Q353
\flag_raise:n Q377
\flat x328
\floatingpenalty L441
\floatpagefraction L278, S2298
\floatsep S726, S744, S751, S2104, S2154, S2303
\flushbottom O55
flushleft (environment) D387
\flushleft D387
flushright (environment) D389
\flushright D389
\fmtname c1, c38, c40, c43, c45, c48, c56, P556, P560
\fmtversion c1, c19, c38, c40, c43, c45, c48, c56, c84, d268, j2, o1498, w160, H1, P147, P157, P573, P576, S4, U492, U518, I1
\fnsymbol *258*
\fnsymbol p107
\font b439, b444, f737, f740, f752, f755, o297, o298, o299, o436, o443, o793, o800, o986, o987, o1034, o1182, r81, r87, r89, t84, w403, w436, w442, w468, y8, y9, y10, z85, A6, A47, A542, A549, D440
\fontdimen b439, b444, o297, o298, o299, o436, o443, o793, o800, w403, w436, w442, z85, A6, A47, A542, A549, I126, I129, I679
\fontencoding *277*, d251, d252, d274, d275, o1529, r251, r282, v269, w508, x16, x24, x67, x74, x83, x85, A36
\fontfamily *277*, *378*, r279, w6, w9, w12, w80, w91, w332, w335, w338, w551, x58, x69, x76, x87, A28, A30, A32, A34, A106, A108, A110, A112, A825
\fontname o987, r89
\fontseries *277*, *303*, *308*, *375*, *377*, *378*, r279, s387, s581, w15, w18, w210, w211, w212, w213, w229, w230, w231, w232, w282, w283, w284, w285, w293, w294, w295, w296, w326, w329, w552
\fontseriesforce *303*, s386, s583
\fontshape . *277*, *308*, o446, o803, r279, s451, s456, s461, s521, s552, s556, s559, s562, s565, s568, s571, s574, s582, w21, w24, w27, w30, w553, A552
\fontshapeforce s522, s584
\fontsize m6, o302, o328, o360, o1184, o1220, r79, r289, w491, w554, A125, A584, A842, L385, L392, L413, L420
\fontsubfuzz t395, t429, D43, D83
\FOO *71*, *71*
\footins L366, L437, S314, S315, S316, S317, S375, S422, S482, S490, S494, S517
\footnote L428
\footnotemark K10, L450
\footnoterule G341, L370, S493
\footnotesep G360, L427, L440, L448
\footnotesize G353, L438
\footnotetext K12, L467
\footskip S77, S637, S696
\forall x323
\frac E341
\frame G138, G214
\framebox *512*
\framebox G161
\frenchspacing b364, f777, f798, n48, n117, n175, D445, D471, D561
\frown x451
\fussy O64
\futurelet e30, f677, f691, l410, l418, z83, E243, H359

G

\Gamma x297
\gamma x270
\gcd E33
gdef commands:
\gdef_ E256
\ge x417, x419
\GenericError j18, j85, j111, j137, t62
\GenericInfo c85, c88, c93, e44, j4, j104, j130, j155, t31, t34, t39, t75, P1437
\GenericWarning j11, j94, j120, j146, t42, t47, t50, t78
\geq x415, x417
\GetFileInfo x3
\gets x439, x441
\gg x433
\globaldefs r564, t185, v60, v89, v149, v180, v212
\glossary *622*
\glossary K182, M23, M35, O24, O32, S621, S680
\glossaryentry M32
\goodbreak b411, f778, f799
\grave x517

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

group commands:
\group_begin: . g141, Q202, Q215, R53
\group_end: ... g144, Q211, Q223, R55
\guillemetleft o537, o767, o1079
\guillemetright o538, o768, o1095
\guillemotleft o540, o770, o1081
\guillemotright o541, o771, o1097
\guilsinglleft o542, o1149
\guilsinglright o543, o1150

H

\H ... j24, o230, o385, o466, o595, o603, o622, o630, o747, o1212, o1351, o1352, o1379, o1380, A188, A205
\halign . *81*, b434, E127, E197, E357, E500
\hangindent K139
\hat x523
\hbadness b316, r618, r625, r660, r679, R162, R164, R214, R216
\hbar x335
\hbox *700*, *701*, *704*
hbox commands:
\hbox:n R297
\hbox_set:Nn . R141, R167, R192, R221
\hbox_set_to_wd:Nnn R165, R217, R253
\hbox_unpack:N R175, R219
\headheight S75, S626, S685
\headsep S76, S635, S694
\heartsuit x333
\height o1188, G31, G34
\hfil *498*
\hfuzz b339, r626, O60, O61, O67, O68, R161, R163, R213, R215
\hglue b401
\hideoutput b499
\hideskip b307, b425
\hidewidth b425, o327, o329, o358, o362, o390, o391, o394, o397, o473, o474, o478, o481, o483, o486, o498, o503, o519, o754, o755, o758, o761, o828, o831, o1219, o1221
\hline H358, H361
\hom E29
hook commands:
\hook_debug_off: *111*, g7, g943
\hook_debug_on: *111*, g7, g942
\hook_gput_code:nnn *109*, *110*, *126*, *128*, g171, g212, g902
\hook_gput_next_code:nn *110*, *128*, g217, g743, g904
\hook_gremove_code:nn *110*, *130*, g286, g906
\hook_gset_rule:nnnn *110*, *132*, g328, g945, g947, g950
\hook_if_empty:nTF *103*, *104*, *110*, g829, g952, R70, R73, R79, R103
\hook_if_empty_p:n *110*, g668, g829, R292
\hook_if_exist:n *107*, *110*
\hook_if_exist:nTF . *103*, *104*, *110*, *130*, g45, g181, g194, g242, g300, g418, g634, g691, g749, g841, g951
\hook_if_exist_p:n .. *110*, g667, g841
\hook_log:n *111*, g616, g941
\hook_new:n *109*, *122*, *148*, *149*, g41, g72, g242, g898, R95, R96, R97, R98, R99
\hook_new_pair:nn *109*, g71, g900
\hook_new_reversed:n *109*, g64, g72, g899
\hook_show:n *111*, *142*, g616, g940
\hook_use:n *103*, *104*, *109*, *113*, *126*, *146*, *148*, g409, g769, g824, g938, R47, R72, R76
\hook_use_once:n *103*, *109*, *113*, *148*, g817, g939
hook internal commands:
\g__hook_..._code_prop g41
\g__hook_??_code_prop g325
\g__hook_??_reversed_tl g325
\g__hook_⟨*hook*⟩_labels_clist ... *123*
\g__hook_⟨*name*⟩_code_prop *122*
\g__hook_⟨*name*⟩_reversed_tl ... *122*
\g__hook_all_seq g28, g48, g398
__hook_apply_-rule_->:nnn ... g591
__hook_apply_-rule_<-:nnn ... g591
__hook_apply_-rule_<:nnn g591
__hook_apply_-rule_>:nnn g591
__hook_apply_-rule_x:nnn g591
__hook_apply_label_pair:nnn *138*, *139*, *145*, g464, g465, g518
__hook_apply_rule:nnn *140*, g528, g534
__hook_apply_rule:nnnN *140*
__hook_apply_rule_->:nnn g569
__hook_apply_rule_<-:nnn g569
__hook_apply_rule_<:nnn g541
__hook_apply_rule_>:nnn g541
__hook_apply_rule_xE:nnn g555
__hook_apply_rule_xW:nnn g555
__hook_clear_next:n g743
__hook_clist_gput:Nn g426, g428, g488, g516
\l__hook_cur_hook_tl *141*, g30, g450, g575, g586
__hook_curr_name_pop: ... g907, g959

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

__hook_curr_name_push:n *151*, g907, g957
__hook_currname_or_default:n *124*, *125*, g90, g105, g121, g122
__hook_debug:n ... *120*, g7, g193, g397, g402, g415, g433, g469, g489, g543, g550, g557, g565, g571, g582
\g__hook_debug_bool . g6, g10, g15, g21
__hook_debug_gset: g7
__hook_debug_label_data:N g469, g510, g604
__hook_debug_print_rules:n .. g727
__hook_declare:n *149*, g50, g56, g206, g335, g748
\g__hook_execute_immediately_-prop *148*, g32, g175, g828
__hook_file_hook_normalize:n *130*, g224, g271, g805
\l__hook_front_tl g441, g480, g483, g486, g488, g489, g490, g503, g504
\c__hook_generics_file_prop *129*, g264, g283
\c__hook_generics_prop ... g240, g281
\c__hook_generics_reversed_ii_-prop g243, g283
\c__hook_generics_reversed_iii_-prop g246, g283
__hook_gput_code:nnn g171
__hook_gput_next_do:nn *128*, g217, g743
__hook_gput_next_do:Nnn g743
__hook_gput_undeclared_hook:nnn *128*, g204, g212
__hook_gremove_code:nn g286
__hook_gremove_code_do:nn *131*, g296, g305
__hook_gset_rule:nnnn g328
\g__hook_hook_curr_name_tl *125*, *151*, *152*, *152*, g34, g124, g130, g920, g927
__hook_hook_gput_code_do:nnn g80, g171, g207
__hook_if_exist:nTF g58, g290, g636, g831, g847
__hook_if_exist_p:n g847
__hook_if_exist_use:n *147*, g794
__hook_if_file_hook:wTF *128*, *129*, *147*, g221, g254, g802
__hook_if_file_hook_p:w g254
__hook_if_label_case:nnnnn g387, g462, g707
__hook_if_marked_removal:nnTF g178, g317
__hook_if_reversed:nTF g424, g674, g676, g853
__hook_if_reversed_p:n g853
__hook_include_legacy_code_-chunk:n g53, g73, g417
__hook_initialize_all: .. g395, g955
__hook_initialize_hook_code:n *139*, g396, g412, g792
__hook_initialize_single:NNNn *134*, *136*, *137*, g430, g446
\l__hook_label_0_tl g441
__hook_label_if_exist_apply:nnnTF *139*, *140*, *145*, g518
__hook_label_ordered:nn *134*
__hook_label_ordered:nnTF *133*, g350, g356, g362, g379
__hook_label_ordered_p:nn ... g379
__hook_label_pair:nn *133*, *134*, g349, g355, g361, g365, g367, g370, g371, g600, g601
\l__hook_labels_int *139*, g441, g449, g453, g485, g506
\l__hook_labels_seq g441, g448, g454, g472, g606
__hook_list_if_rule_exists:nnnTF g700
__hook_list_one_rule:nnn g700
__hook_list_rules:nn *144*, g654, g700, g732
__hook_log:nN g616
__hook_log_cmd:n g618, g623, g627, g629, g633
__hook_log_line:n g616
__hook_log_line_indent:n g616
__hook_log_next_code:n .. g650, g698
__hook_make_name:n *122*, *126*, g115, g121, g127, g132
__hook_make_name:w g132
__hook_mark_removal:nn *131*, g297, g303, g307
__hook_msg_pair_found:nnn g543, g550, g557, g565, g573, g584, g597
\g__hook_name_stack_seq g34, g910, g916, g919, g925, g926, g935
__hook_new:n g42, g43, g68
__hook_new_reversed:n g64
__hook_normalize_hook_args:Nn .. g42, g65, g139, g619, g624, g744, g822
__hook_normalize_hook_args:Nnn g139, g172, g287

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

__hook_normalize_hook_args_-aux:Nn g139
__hook_normalize_hook_rule_-args:Nnnnn g139, g330
__hook_parse_dot_label:nn . g93, g96
__hook_parse_dot_label:nw g96
__hook_parse_dot_label_aux:nw . g96
__hook_parse_dot_label_cleanup:w g96
__hook_parse_label_default:n . . . g87, g151, g157, g158, g165, g166, g168
__hook_preamble_hook:n *146*, *147*, g410, g632, g769, g813
\l__hook_rear_tl g441, g470, g476, g477, g499, g500
\g__hook_removal_list_prop g29
\g__hook_removal_list_tl *132*, g29, g309, g314, g319
__hook_removal_tl:nn g310, g315, g320, g323
\l__hook_return_tl g25, g197, g200, g295, g486, g487, g821, g823, g925, g926, g927, g935
__hook_rule_<_gset:nnn g347
__hook_rule_>_gset:nnn g347
__hook_rule_after_gset:nnn . . g347
__hook_rule_before_gset:nnn *138*, g347
__hook_rule_gclear:nnn *134*, g336, g368
__hook_rule_incompatible-error_-gset:nnn g364
__hook_rule_incompatible-warning_-gset:nnn g364
__hook_rule_unrelated_gset:nnn *134*, g368
__hook_rule_voids_gset:nnn . . g359
__hook_seq_csname:n g439, g456, g490, g546, g553, g611
__hook_str_compare:nn g23, g373, g381, g390
__hook_strip_double_slash:n . g271
__hook_strip_double_slash:w . g271
__hook_tl_csname:n g439, g445, g455, g471, g474, g476, g480, g492, g494, g497, g499, g504, g544, g545, g551, g552, g610
__hook_tl_gput:Nn g425, g427, g487, g516
__hook_tmp:w g36, g702, g723, g730, g741, g907, g911, g913
\l__hook_tmpa_bool *143*, g24, g653, g656, g664, g673
\l__hook_tmpa_tl g25
\l__hook_tmpb_tl g25
__hook_try_declaring_generic_-hook:nnn *128*, g186, g209
__hook_try_declaring_generic_-hook:nNNnn *128*, *128*, g211, g216, g219
__hook_try_declaring_generic_-hook:wnTF g219
__hook_try_declaring_generic_-hook_split:nNNnn g219
__hook_try_declaring_generic_-next_hook:nn *128*, g209, g751
__hook_try_file_hook:n . . . *147*, g794
__hook_unmark_removal:nn *127*, *131*, g179, g312
__hook_update_hook_code:n *127*, *131*, *135*, g184, g301, g341, g394, g396, g400, g762
__hook_use:wn *146*, *147*, g776, g789, g794
__hook_use_end: g769
__hook_use_initialized:n *146*, g409, g769
__hook_use_once_store:n . g822, g827
__hook_use_undefined:w g769
\g__hook_used_prop g33, g397, g404, g434
\l__hook_work_prop *138*, g31, g429, g451, g458, g460, g469, g486, g510, g578, g589
\hookleftarrow x480
\hookrightarrow x478
\hphantom E75
\hrule b402, b446, l350, l358, l384, l392, o289, o294, x340, x618, G144, G149, G197, G207, H359, H376, L371, I590, I600
\hrulefill b446, f779, f800
\hspace l437, l441, l446
\hss *713*
\Hwithstroke *232*, o491, o1179
\hwithstroke *232*, o507, o1180
\hyphenation o205
\hyphenchar *98*, f737, f740, f749, f752, f755, f760, D440
\hyphenpenalty b319, r632, r664

I

\I b370, P1047, P1178, P1268, P1288, U188, U453
\i o247, o402, o449, o450, o451, o452, o453, o454, o544, o581, o582, o674, o676, o678, o680, o772, o1115, o1261,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

o1263, o1265, o1267, o1318, o1321, o1324, o1327, o1397, U192, U457, U464
\ialign b434, b436, x337, x459, x530, x533, x537, x540, E155, E157, E176, H191, I141
\IeC U255, U259, U366
\if *465*
if commands:
\if_case:w g373, g390
\if_cs_exist:w g527, g722, g780
\if_int_compare:w g381, g855
\if_meaning:w Q232
\IfBold *380*
\IfClassAtLeastTF P143
\ifcsname . f635, f652, s403, s406, s537, s540, w39, w50, P935, P1066, R347
\ifdefined e52, e53, e54, e55, e91, e106, e107, e108, e124, w473
\iff x500
\IfFileExists *193*, *637*
\IfFileExists *210*, a178, e18, e49, e93, e103, n407, n447, n478, n534, n559, P685, Q146, Q178, Q188, U486
\iffontchar o1034, o1182
\IfFontSeriesContextTF *380*, w346, w382, w384
\IfFormatAtLeastTF P143
\IfHookEmptyTF *103*, *104*, *106*, *153*, g951, g988, D199
\IfHookExistsTF *103*, *104*, *107*, g951, g987
\ifinner E264, E272, E292, E319, L57, L126, L315
\ifnum *703*
\ifodd v973, L68, L137, S21, S138, S610, S668, S982, S985, S1018, S1021, S1132, S1135, S1294, S1297, S1574, S1577, S1695, S1698, S1818, S2039, S2047, I320, I344, I378, I400
\IfPackageAtLeastTF P143
\IfTargetDateBefore P1496
\ifvbox S319, S376, S423, S502, S518
\ifx *96*, *370*, *371*, *372*, *384*, *691*
\ignorespaces *117*, *220*, l49, l143, l162, l174, l185, l201, l214, l483, n73, n140, n195, o72, r286, D205, D212, D263, D278, D323, D328, D334, E300, E327, F55, F217, G135, G360, H57, H62, H68, H83, H92, H105, H109, H116, H123, H125, H134, H154, H237, H301, H303, H305, H332, J30, J32, K110, L17, L24, L448, N7, N9, R254, I36, I46, I66, I75, I109, I120, I131, I143, I148, I154
\ignorespacesafterend D7
\IJ o250, o433, o547, o1116
\ij o249, o431, o546, o1117
\Im x314
\imath x309
\in x429, x461
in commands:
in_callback d791
\in_callback *41*
\include *193*
\include *119*, *204*, *679*, *681*, *682*, *685*, n219, n267, n269, n285, n287
include/after *682*
include/after/... . . . . *682*
include/before *682*
include/before/... . . . . *682*
include/end *682*
include/end/... . . . . *682*
\IncludeInRelease *68*, a18, a23, a290, a307, b49, b88, b103, b119, b125, b134, b139, b148, b154, b168, b182, b186, b220, b228, b233, b244, b289, b457, b492, b499, b546, c66, d3, d227, d250, d273, e2, e39, e101, e121, e129, e141, f5, f13, f56, f62, f272, f314, f342, f370, f377, f394, f410, f448, f460, f488, f499, f511, f518, f614, f631, f661, f702, f721, f732, f749, f760, f771, f791, g3, g963, l5, l21, l54, l65, l81, l86, l98, l104, l132, l152, l166, l178, l191, l206, l235, l252, l271, l307, l333, l366, l426, l432, l440, l445, l459, l465, n12, n85, n143, n221, n258, n279, n294, n341, n402, n438, n484, n489, n497, n511, n522, n538, n571, o77, o104, o148, o169, o324, o332, o353, o369, p24, p30, p46, p90, p127, p143, p151, p169, q5, q11, r24, r53, r210, r232, r391, r402, r410, r442, r450, r464, r485, r521, r608, r670, s4, s578, s611, s617, s621, t495, t504, u2, u22, v49, v78, v138, v169, v200, v607, v655, v701, v710, v894, v905, w33, w163, w188, w267, w319, w348, w381, w393, w427, w440, w471, w480, w505, w518, x63, x81, x100, x110, x622, x634, z27, z34, A2, A42, A59, A533, C38, C51, C62, D10, D66, D113, D140, D159, D215, D265, D287, D295, D308, D314, D326, D331, D341, D360, D377, D401, D423, D450, D475, D488,

D503, D519, D542, D552, E79, E87, E109, E118, E139, E144, E152, E163, E203, E220, E259, E267, E277, E304, E385, E394, E431, E443, E455, E464, F125, F133, G4, G14, G50, G70, G98, G106, G162, G170, G216, G225, G262, G284, G364, G370, G390, G398, H60, H65, H137, H157, H222, H227, K5, K20, K161, K168, K174, K187, K210, L35, L105, L206, L232, L280, L294, L382, L388, L396, L403, L410, L416, L422, O20, O40, P17, P22, P37, P53, P65, P76, P84, P90, P108, P117, P145, P155, P167, P180, P210, P229, P249, P265, P279, P297, P310, P326, P346, P364, P432, P450, P474, P486, P527, P537, P593, P622, P654, P782, P862, P869, P923, P1055, P1185, Q5, Q15, Q27, Q88, Q141, Q174, Q185, Q197, Q240, Q248, Q260, Q281, Q294, Q301, Q324, Q344, Q369, Q415, Q427, Q438, R3, R344, R391, R405, S24, S54, S151, S179, S345, S366, S371, S419, S591, S651, S794, S812, S873, S894, S930, S954, S1066, S1217, S1386, S1468, S1562, S1684, S1903, S1930, S2160, S2178, S2197, S2243, U8, U16, U23, U38, U57, U66, U73, U99, U132, U221, U226, U246, U339, I10, I16, I26, I39, I55, I69, I80, I89, I100, I112, I146, I151, I160, I172, I235, I250, I311, I369, I455, I464, I473, I501, I531, I558, I572, I585, I596, I608, I625, I645, I652, I686, I746, I804, I820

`\includeonly` *193*

`\includeonly` *201*, *681*, *682*, n219, n259, n261, n280, n281

`\indent` l462, F161, H81

`\index` *622*

`\index` K182, M6, M18, O24, O32, S620, S679

`\indexentry` M15

`\inf` E25

`\infty` x316

`\initcatcodetable` d91

`\input` *193*, *637*

`\input` *119*, *634*, *679*, *680*, *680*, *684*, *685*, *691*, a68, a174, a177, a234, d18, e90, e94, e119, f22, n494, t16, u106, w564, w574, w584, x10, x11, x12, x13, x14, x23, x41, x42, x46, x47, x136, x137, x138, x139, x654, x655, x656, A1038, P503, U97, U111, U136, U214, U303, U491

`\input@path` *1*, *6*

input@path commands:

`\input@path:` a239

`\inputencodingname` . . . U280, U302, U384

`\InputIfFileExists` *193*, *637*

`\InputIfFileExists` *209*, *210*, *656*, *658*, *685*, *687*, *687*, *688*, n477, n500, n515, n525, n535, n589, o1506, r362, w556, w566, w576, A754, A1121, P751, P813, Q140, T8, U208

`\inputlineno` a327, j165

`\insert` b254, b279, b281, b284, b299, L437, S517, S518, S1883

`\int` x348

int commands:

`\int_compare:nNnTF` . . . g474, g496, g506, g865, R34, R78, R277, R283

`\int_decr:N` g485

`\int_eval:n` . . . g493, g545, g552, R279

`\int_gdecr:N` Q486

`\int_gincr:N` Q476, R48, R64

`\int_gset:Nn` Q472

`\int_incr:N` g453

`\int_new:N` . . . g442, Q471, R270, R272

`\int_set:Nn` Q204, Q217

`\int_use:N` *703*, *703*, Q480, R66, R81, R287

`\int_value:w` R27

`\int_zero:N` g449

`\c_max_int` . . . R138, R164, R189, R216

`\c_zero_int` R52

int internal commands:

`\__hook~` *126*

`\__hook~...` g41

`\__hook~??` *132*, g325

`\__hook` ⟨*hook*⟩ *136*

`\__hook` ⟨*name*⟩ *122*

`\interdisplaylinepenalty` l13, E55, E194, E379

`\interfootlinepenalty` b360

`\interfootnotelinepenalty` b360, l18, L439

`\interlinepenalty` l11, r634, D412, D415, D433, D436, K67, K118, K194, K217, L439, S338, S1153, S1157, S1319, S1323

`\intextsep` S1136, S1140, S1155, S1158, S1165, S1298, S1304, S1321, S1324, S1333, S2303

`\intop` x347, x348

`\iota` x276

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

iow commands:
\iow_char:N g575, g586
\iow_log:n g618
\iow_newline: Q484
\iow_now:Nn R286
\iow_term:n g193, g403, g405, g415, g489, g508, g509, g511, g574, g585, g599, g605, g606, g607, g610, g614, g623, g729, g734, Q477, Q510
\ishortstack I132
\itdefault s565, w30, x94
\item *498*, j230, D337, D387, D389, D403, D425, E438, E450, E477, F141, F219, H78, J36, J38, N4, N8
\itemindent *497, 498*, F9, F42, F95, F187, F208
itemize (environment) F242
\itemize F242
\itemsep *498*, F1, F176
\iterate a81, a82, b390
\itshape *309*, *382*, o444, o801, s563, s564, w28, w29, w404, w437, w443, z21, A550, J36, J38, L375

J

\J U190, U455
\j o248, o403, o545, o773, o1125, o1331, o1411, U464
\jmath x310
\jobname *664*, *688*
\Join w538
\joinrel . x471, x478, x480, x482, x484, x486, x488, x490, x492, x496, x498
\jot E53, E191, E390, E400

K

\k *238*, o482, o584, o589, o611, o616, o692, o693, o751, o752, o806, o808, o813, o815, o1217, o1285, o1286, o1303, o1304, o1326, o1327, o1328, o1381, o1382, o1409, o1410, A190, A208
\kanjiskip e55
\kappa x277
\ker E27
\kern *711*
kernel internal commands:
__kernel_file_name_sanitize:n . Q97
__kernel_msg_expandable_error:nnnn Q400
__kernel_msg_new:nnn Q406
\kill H154, H162

L

\L o242, o421, o527, o765, o1118, P1044, P1175, P1265, P1287, U483
\l . . . o251, o423, o548, o774, o1119, U483
\label . . C32, K182, O24, O32, S619, S678
\labelenumi *509*
\labelenumiv *509*
\labelformat *456*
\labelformat . . . C46, C52, C58, C63, C69
\labelitemi *509*
\labelitemii *509*
\labelitemiii *509*
\labelitemiv *509*
\labelsep *498*, *509*, F9, F210, F216, J36, J38
\labelwidth *496*, *498*, F9, F93, F209, F211, F214
\Lambda x300
\lambda x278
\land x368, x370
\langle x594
\language b35, b82, b84, b99, n54, n123, D408, D548, S597, T10
\lastbox r652, E180, E181, F130, F136, F185, K99, K132, S305
\LastDeclaredEncoding . . r137, r140, U380
\lastnamedcs f653
\lastnodetype r645, r646, r647, r651
\lastpenalty r648, z112, z115
\lastskip b414, b415, b417, b419, l44, l128, l140, l159, l220, l221, l225, l227, l228, l240, l258, l280, l283, l315, l318, l319, z102, z105, F115, F116, F150, F151, I123
\LaTeX f732, m3, m15, P1007, P1138, P1227, R312, R318
\LaTeXe m13
\latexreleaseversion c1
\lbrace o308, x598
\lbrack b374
\lccode j19, j20, j21, j22, j23, j24, o140, o1039, D493, D508, D523, D566, U157, U174, U182, U189, U191, U192, U194, U196, U197, U198, U199, U439, U447, U454, U456, U457, U459, U461
\lceil x602
\ldotp x501, x504, x619
\ldots o322, x505
\le x416, x418
\leaders b446, x340, x558, x559, x561, x562, H376, K199, K222, I565, I578, I590, I600
\leadsto w541

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\leavevmode b405, b432, b435, b446, b448, l407, l421, o75, o184, o289, o290, o393, o422, o426, o429, o476, o757, o790, z123, A119, A836, D412, D433, D446, D457, D465, D544, D554, D567, E438, E450, E477, F58, F103, G8, G17, G24, G137, G139, G155, G183, G244, G320, G377, G394, G401, H178, K40, K195, K207, K218, L462, N14, S157, S162, S184, S189, I134, I314, I373
\left x625, x627, x629, x631, x636, x637, x638, x639, E154, E160, E182
\Leftarrow x410, x492, x498
\leftarrow x437, x439, x480, x490, x496, x550
\leftarrowfill x534, x550
\lefteqn E414
\leftharpoondown x453, x467
\leftharpoonup x452
\lefthyphenmin T11
\leftline G428
\leftmargin *496*, *498*, F9, F52, F53, F94, F146, F148
\leftmargini *498*, E430, F17
\leftmarginii F17
\leftmarginiii F17
\leftmarginiv F17
\leftmarginv F17
\leftmarginvi *498*, F17
\leftmark O48
\Leftrightarrow x409
\leftrightarrow x436
\leftskip b427, r629, D345, D351, D355, D365, D369, D373, D405, D427, F74, G276, G297, K192, K197, K215, K220
\legacyoldstylenums A1, A52
\leq x414, x416
\let ... *83*, *86*, *153*, *269*, *403*, *403*, *407*, *718*
\LetLtxMacro *90*
\lfloor x606
\lg E4
\lgroup x608
\lhd w544
\lhook x477, x478
\lim E6
\liminf E8
\limits x539, x543, E149, E340
\limsup E7
\line j219, I158, I450, I809, I826
\linebreak *173*
\linebreak l9, l26
\linepenalty b318
\lineskip b368, b400, b435, x458, E187, G278, G298, H71, H198, R147, R198, S622, S681, I136, I315, I374
\lineskiplimit b369, b400, b437, b438, x458, x510, E189, E193, G264, G279, G286, R148, R199, S622, S681
\linespread r287
\linethickness I130, I810, I827
\linewidth *497*, n32, n101, n159, E285, E311, E439, E451, E478, E482, E500, F15, F51, F52, F54, G274, G295, H36, L266, S146, S205
list (environment) F34
\list F34, F236, F247
\listfiles *637*
\listfiles *118*, n604
\listparindent *498*, F9, F41, F50
\ll x434
\llap F238, F249, G432
\lmoustache x563
\ln E5
\lnot x326, x327
\LoadClass *635*
\LoadClass P509, P523, P719, P830, P904, P912, P913
\LoadClassWithOptions *635*
\LoadClassWithOptions P522
\LoadFontDefinitionFile r389, r415, r416, x21, x27, x28, x29, x33
\LoadPackageWithOptions *680*
\loccount d17
\log E3
\loggingall b457
\loggingoutput ... b453, b466, b482, b496
\LogHook *108*, g940
\long *82*
\Longleftarrow x492
\longleftarrow x489
\Longleftrightarrow x498, x500
\longleftrightarrow x496
\longmapsto x494
\Longrightarrow x486
\longrightarrow x487, x494
\loop a81, b390, d150, d159, r643, H382, P953, P1016, P1084, P1147, P1198, P1236, U263, U274, U284, U295, U325, U351, U361
\lor x369, x371
\lower m2, x458, G195, I35, I45, I196, I305, I306, I353, I354, I409, I410
\lowercase j26, o141, o1040, o1504, r303, r361, D497, D512, D527, D567, U479
\lq b372
lrbox (environment) *512*

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\lrbox G125
\ltfilehookdate Q469
\ltfilehookversion Q469
\luabytecode d194
\luachunk d202
\luafunction *38*, d178
luatexbase d280
\luatexluafunction a18, a23
\luatexversion . . a11, d5, e54, e108, o977

M

\M b370
\magstep b361
\magstephalf b361
\makeatletter . . . . . .
. . . . f728, n34, n103, n161, r367,
D25, D80, K151, P503, P683, P791, S2
\makeatother f728, P503, U546
\makebox *512*
\makebox E285, E311, G3
\makeglossary *622*
\makeglossary n206, M20
\makeindex *622*
\makeindex n205, M3
\makelabel . . . . . .
497, F45, F97, F205, F218, F238, F249
\MakeLowercase U473, U485
\MakeRobust *362*,
f271, f773, f774, f775, f776, f777,
f778, f779, f780, f781, f782, f783,
f784, f785, f786, f787, f788, v705,
v900, I806, I807, I808, I809, I810,
I811, I812, I813, I814, I815, I816, I817
\maketitle *588*
\MakeUppercase C48, C60, U463
\mapsto x444
\mapstochar x443, x444, x494
\marginpar L308
\marginparpush S85, S1834
\marginparsep S84, S1845, S1847
\marginparwidth L337, S83, S1847
\mark O27, O35, O53
\markboth O21, O22, O41, O43
\markright O22, O44
\marks d37, U10, U12
math (environment) E332
\math E332
\mathaccent v622, v670, v704, v714
\mathalpha v792,
v971, x169, x170, x171, x172, x173,
x174, x175, x176, x177, x178, x179,
x180, x181, x182, x183, x184, x185,
x186, x187, x188, x189, x190, x191,
x192, x193, x194, x195, x196, x197,
x198, x199, x200, x201, x202, x203,
x204, x205, x206, x207, x208, x209,
x210, x211, x212, x213, x214, x215,
x216, x217, x218, x219, x220, x221,
x222, x223, x224, x225, x226, x227,
x228, x229, x230, x297, x298, x299,
x300, x301, x302, x303, x304, x305,
x306, x307, x516, x517, x518, x519,
x520, x521, x522, x523, x525, x528
\mathbf w14, w199, w280, w325, x151
\mathbin v976, x232, x233, x235,
x358, x359, x360, x361, x364, x365,
x366, x367, x370, x371, x372, x373,
x374, x375, x376, x377, x378, x379,
x380, x381, x382, x383, x384, x385,
x386, x387, x388, x389, x390, x391,
x392, x393, x394, x395, x396, x397, E37
\mathcal x150
\mathchar b433, v733, v777, x335, x336, x617
\mathchardef . . . . . .
. . b21, b22, b23, b24, b107, b110,
b111, d219, h3, h4, h5, h6, o70, v768
\mathcharzero d219
\mathchoice E61
\mathclose . . . . . .
v979, x231, x240, x242, x245, x250,
x256, x258, x260, x566, x593, x597,
x601, x605, x611, E43, E46, E49, E52
\mathcode v765, x252, x253, x254
\mathdollar o307, x614
\mathellipsis o321, x619
\mathgroup . b79, r14, t257, t263, t269,
t270, t281, x643, A8, A14, A49, A1048
\mathhexbox b433, w502
\mathindent E428, E440, E452, E480, E490
\mathinner x504, x508, x513, x619
\mathit s564, w29, x153, x156, x617
\mathnormal x149
\mathop v975, x341, x342, x343,
x344, x345, x346, x347, x349, x350,
x351, x352, x353, x354, x356, x357,
x537, x540, E3, E4, E5, E6, E7, E8,
E9, E10, E11, E12, E13, E14, E15,
E16, E17, E18, E19, E20, E21, E22,
E23, E24, E25, E26, E27, E28, E29,
E30, E31, E32, E33, E34, E149, E340
\mathopen v978, x241, x244, x249,
x255, x257, x259, x564, x595, x599,
x603, x607, x609, E41, E44, E47, E50
\mathord . . . . . .
v792, v974, x236, x243, x246, x251,
x263, x264, x265, x267, x268, x269,
x270, x271, x272, x273, x274, x275,
x276, x277, x278, x279, x280, x281,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

x282, x283, x284, x285, x286, x287, x288, x289, x290, x291, x292, x293, x294, x295, x296, x308, x309, x310, x311, x312, x313, x314, x315, x316, x317, x318, x319, x320, x321, x322, x323, x324, x325, x327, x328, x329, x330, x331, x332, x333, x334, x524, x526, x527, x549, x550, x553, x554, x555, x556, x568, x570, x572, x575, x577, x591, x613, x614, x615, x616
\mathpalette x457, x461, x464, E60, E69, E99, E129
\mathparagraph . . . o310, p134, p146, x614
\mathpunct v980, x234, x238, x501, x502, x503
\mathrel v977, x237, x239, x247, x248, x261, x262, x338, x398, x399, x400, x401, x402, x403, x404, x405, x406, x407, x408, x409, x410, x411, x414, x415, x418, x419, x420, x421, x422, x423, x424, x425, x426, x427, x428, x429, x430, x432, x433, x434, x435, x436, x437, x438, x441, x442, x443, x445, x446, x447, x448, x449, x450, x451, x452, x453, x454, x455, x457, x461, x464, x471, x473, x476, x477, x479, x482, x484, x579, x581, x583, x585, x587, x589, E42, E45, E48, E51, E149, E340
\mathring x528
\mathrm w5, w238, w302, w331, x148
\mathsection o311, p133, p145, x614
\mathsf . w8, w243, w307, w334, x152, x155
\mathsterling o319, x614
\mathstrut E84, E93, E158, E159
\mathsurround b421
\mathsymbol v770
\mathtt w11, w248, w312, w337, x154
\mathunderscore x614
\mathversion r307, w464, w466
\matrix E156, E160, E167
\max E22
\maxdeadcycles S7
\maxdepth b342, l287, n62, n130, n185, S92, S169, S170, S506, S514, S546, S715, S724, S764, S991, U85
\maxdimen *716*, b307, b343, b344, b400, b438, b454, b465, b481, b496, r616, r626, r661, r676, t338, t391, x458, P1308, P1349, P1358, R275, R277, R324, S291, S1853, S1873, S1878, S2165, S2205, S2206, S2208, U89, I475, I503, I533, I611, I628
\mbox *512*
\mbox b433, m13, o293, w498, x506, G11, G20, G24, L385, L392, L413, L420, I52
\mddefault *368*, *368*, *369*, *371*, *373*, w18, w195, w221, w222, w223, w232, w275, w296, w329, x92, x105, x107, x121
\mdseries *119*, *373*, *380*, w16, w17, w156, w218, w290, w291, w327, w328, w501, z20
mdseries w252
mdseries/defaults w252
\meaning a219, a228, a323, f228, f290, f328, f356, f439, f699, v444, v457, v558, v623, v670, v734, v828, v924, v1028
\medbreak b415, f780, f801
\mediumseries *373*
\medmuskip x645, E36, E38, E211, E214, E228
\medskip b418, l400
\medskipamount b417, l401, l403
\medspace E201
\MessageBreak e59, e62, e63, e64, e65, e66, e67, e80, e81, e82, e83, e84, f204, f279, f318, f346, j3, j6, j13, j33, j46, j60, j73, j171, j173, j179, j186, o161, o971, o1509, o1512, r34, r35, r507, r541, s419, t20, t21, t67, t88, t281, t432, t452, t484, t500, t515, t528, u31, u33, v399, v408, v546, w58, z144, A23, A100, A102, A121, A760, A762, A763, A764, A766, A768, A769, A770, A771, A772, A822, A824, A831, A838, A1053, D44, D84, P257, P271, P549, P560, P562, P564, P575, P671, P672, P674, P675, P676, P678, P680, P714, P715, P716, P717, P798, P799, P801, P802, P803, P805, P807, P825, P826, P827, P828, P889, P906, P907, P975, P994, P1033, P1106, P1125, P1164, P1219, P1253, P1362, P1364, P1446, P1449, P1462, P1464, R60, R108, R398, R399, R400, R401, S578, S1960, S1997, U232, U233, U234, U236
\mho w537
\mid x402
\min E23
minipage (environment) *513*
\minipage G307
\mit w589
\mkern x335, x338, x340, x462, x471, x513, x514, x515, x545, x546,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

x547, x548, x549, x550, x551, x552, E36, E37, E40, E73, E74, K200, K223
mlist commands:
mlist_to_hlist d840
\models x484
module commands:
module_error d336
module_info d336
module_warning d336
\module_error *41*
\module_info *41*
\module_warning *41*
modules d289
\month a185, c17, P1011, P1142, P1231
\moveright S625, S684
\mp x389
\mscount H379
msg commands:
\msg_error:nnn g46, g929, Q74
\msg_error:nnnnnn g343, g558
\msg_expandable_error:nnn g100
\msg_line_context: g877
\msg_new:nnn g876, g893
\msg_new:nnnn g861, g871, g878
\msg_warning:nnnnnn g566
\mskip E36, E38, E206, E225, E228, E230, E231
\mu x279
\mubyte U250
\multicolumn *469*, H233
\multiput I78, I811, I828
\multispan H233, H379
\muskip b29, b55, b93, d34, x545, x546
\muskipdef b55, b93, d220
\muskipzero d220

N

\n d323, d325, d332, d334, d461, d567, d593, d619, d665, d687, d706, d714, d715, d735, d748, d755, d756, d763, d775
\nabla x319
\NAME *71*
\narrower b426
\natural x329
\ncallback d670
\ndefault d675, d679
\ne *403*, x413
\nearrow x405
\NeedsTeXFormat . . t12, A731, P554, P1508
\neg x325, x326
\negmedspace E201
\negthickspace E201
\negthinspace l468, E201
\neq *403*, x412
new commands:
new_attribute d402
new_bytecode d436
new_chunkname d449
new_luafunction d465
new_whatsit d424
\new_attribute *39*
\new_bytecode *39*
\new_chunkname *39*
\new_luafunction *39*
\new_whatsit *39*
\newattribute *38*
\newattribute d74, d230
\newbox b47, b312, b423, h13, D468, E66, F27, G96, H16, H17, H18, H343, S86, S120, S121, S122, I6, I670, I675
\newcatcodetable *38*
\newcatcodetable d84, d93, d94, d120, d121, d234
\newcommand *71*
\newcommand *83*, *86*, *90*, *92*, f77, o4, s453, s458, s463, w36, x51, x52, x53, x54, x56, x57, x59, x60, x92, x93, x94, x95, x96, x97, x120, x121, x122, D289, D290, D291, D292, Q500, Q501, Q502, Q503, Q505, S2291, S2294, S2297, S2298, S2301, S2302, I682
\NewCommandCopy *87*, *88*, *89*, f461, f463
\newcount b47, b360, h7, h8, l124, n9, p36, t25, v27, v286, E55, E344, E345, F23, F24, F25, F26, F56, F226, F241, G348, H11, H12, H13, H14, H15, H335, H336, H337, K36, K140, K141, L3, L267, L268, L269, L270, P1305, S103, S105, S107, S109, S111, S119, S1986, S2289, S2292, S2295, S2299, U3, U4, U5, U77, I664, I665, I666, I667, I676
\newcounter *258*
\newcounter p10
\newdimen b47, b307, b309, b310, b359, h10, h11, h12, l123, t352, t353, E53, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21, F22, G152, G153, H3, H5, H6, H7, H8, H166, H338, H339, H340, H341, L427, S71, S72, S73, S75, S76, S77, S78, S79, S80, S81, S82, S83, S84, S85, S91, S93, S94, S106, S108, S110, S112, S113, S114, S115, S116, S117, S118, S1987, S1988, I3, I4,

I5, I7, I431, I432, I433, I434, I435, I436, I668, I669, I671, I672, I673, I674
\NewDocumentCommand *67*, *86*, e91, e122, e124, g898, g899, g900, g901, g903, g905, g933, g944, g946, g949
\newenvironment *72*
\newenvironment *115*, *116*, f146, P1009, P1140, P1229
\newfam b47, d38, r16
\newfont w468
\newgroup v47
\newhelp b306
\NewHook *100*, *101*, *101*, *106*, *107*, *110*, *112*, *116*, g4, g898, g964, g966, n74, n75, n76, w252, w253, w254, w255, w256, w257, w258, w259, w260, D32, D33, D34, D35, D36
\newif c68, f168, f767, h9, n7, n8, r204, s385, v15, w378, z82, A780, C3, E75, E76, E190, E346, F28, F29, F30, F31, F32, F33, F138, G362, H19, H251, K38, K124, P2, S95, S96, S97, S98, S99, S100, S101, S102, I157, I427, I428, I429, I430, I459, I460
\newinsert b193, b242, G349, L366, S27, S1852
\newlabel C22, C34
\newlanguage b47, U217
\newlength *266*
\newlength q3
\newline l92, l99, l105
\newlinechar a72, f20
\newluabytecode *38*
\newluabytecode d189, d244
\newluachunkname *39*
\newluachunkname d197, d246
\newluafunction *38*
\newluafunction d4, d173, d228, d240, o993
\newmarks U6
\newmathalphabet u13, u109
\NewMirroredHookPair *101*, *107*, *110*, g898, g968
\newmuskip b47
\newpage S133, S139, S150
\newread b47, n3
\NewReversedHook *101*, *101*, *107*, *110*, *112*, g898, g967
\newrobustcmd *86*
\newsavebox *512*
\newsavebox G96
\newskip b47, b308, b311, b357, b358, h14, h15, h17, l403, l404, l405, l454, q3, D386, E347, E429, F2, F3, F4, F5, F6, F7, F8, S2303, S2304, S2305, S2309, S2310, S2313, S2314, S2315, S2319, S2320, S2321
\newtheorem J1
\newtie *429*, o845, A194, A212, A615, A924, A925
\newtoks b47, b306, h16, r317, r318, s442, t201
\newwhatsit *38*
\newwhatsit d181, d242
\newwrite b47, n4, n5, n6, K154, M4, M21
\newXeTeXintercharclass U21
next ... internal commands:
__hook_next~... . . . . g41
next ⟨*name*⟩ internal commands:
__hook_next ⟨*name*⟩ *122*
\NG o528, o1120, U483
\ng o549, o1121, U483
\ni x430, x431
\noalign *469*, x339, x531, x534, x537, x538, x542, x543, E158, E159, E175, E178, E192, E390, E400, H224, H230, H359, H378, I148, I154
\nobreak b403, b406, b408, f781, f802, l59, l71, l115, l141, l147, l160, l173, l199, l351, l359, l385, l393, l414, l421, l452, n204, n216, o432, o434, D311, D318, G427, K90, K197, K198, K202, K220, K221, K225, L463, O29, O37, S336, S1149, S1315, U141, U143, U147, U148, U149, U153
\nobreakdashes l406
\nobreakspace l420
\nobreakspace␣ *189*
\nocite *625*
\nocite N39
\nocorr z43, z58, z62, z65
\nocorrlist z89, z121
\nofiles *193*
\nofiles *463*, *464*, *715*, n200
\noindent *506*, r639, r665, K139
\nointerlineskip b398, x339, x531, x534, x538, x542, E284, E310, S1842, S1850, I563, I566, I576, I578
\nolimits x348, x355, E3, E4, E5, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E26, E27, E28, E29, E31, E34
\nolinebreak *173*
\nolinebreak l9, l27
\nonfrenchspacing b364, b552, f782, f803, n50, n119, n177
\nonscript E36, E38
\nonumber E373, E412, E413

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\nopagebreak *173*
\nopagebreak l7, l25
\noprotrusion K207, K230
\normalbaselines b368, E154, E156
\normalbaselineskip b357, b369, t142, G280, G299
\normalcolor *790*, E339, E425, G89, G340, K203, K226, L97, L166, S216, S492, S629, S639, S688, S698, S2226, S2259
\normalfont *119*, *220*, *373*, r617, r677, w503, w519, w521, w527, z18, D447, E339, E425, K203, K226, L377
normalfont w252
\normallineskip . b357, b368, G278, G298
\normallineskiplimit b357, b369, E193, G263, G279, G285
\normalmarginpar L363
\normalsfcodes n46, n48, n50, n115, n117, n119, n173, n175, n177, n199, S618, S677
\normalshape . . . *306*, *309*, s550, s597, s604
\normalsize *102*, n44, n113, n171, z142, L23, L176, L348, P5, S617, S676
\not x338, x412, x413, x435
\notin x461
\nu x280
\null b382, o327, o363, o483, o486, o828, o831, o1219, C17, D412, D433, D544, D554, E112, E121, E156, E185, K197, K220, R300
\nulldelimiterspace b346, x642
\nullfont D156
\number . . a86, c48, c56, d105, f2, f114, p108, r567, r570, t393, v64, v93, v113, v128, v153, v184, v216, w495, P920, P1011, P1142, P1231, R275
\numberline K72, K82, K233, L17
\numexpr b189, b205, b215, b246, d82, d105, d157, f642, o1018, S36
\nunknown d692
\nwarrow x407

O

\O . . . o244, o400, o530, o764, o1105, U482
\o . . . o253, o405, o551, o775, o1111, U482
\oalign b435
\obeycr l480
\obeylines b385, f783, f804, D418, D439, D535, D536, S583
\obeyspaces b385, f784, f805, S583
\oddsidemargin S72, S74, S611, S670
\odot x384
\OE . . o243, o399, o529, o763, o1122, U482
\oe . . o252, o404, o550, o776, o1123, U482
\of E67, E343
\offinterlineskip b398
\oint x355
\ointop x354, x355
\oldstylenums *223*, *423*, *432*, A1, A43, A45, A303, A304, A305, A306, A307, A308, A309, A310, A311, A312, A1045
\Omega x307
\omega x290
\ominus x387
\omit E178, E179, H369, H372, H379, H383
\onecolumn S141
\OnlyDescription t5, y3
\ooalign b435, o327, o357, o394, o477, o483, o485, o496, o512, o725, o758, o828, o831, o877, o1219, w500, x462, x465
\openin *679*
\openup E186, E191
\oplus x388
\OptionNotUsed P391, P398, P844
or commands:
\or: g375, g391
\oslash x385
\OT o372
\otimes x386
\outer d21, d38
\outerparskip F1
\output S256
\outputpenalty S258, S272, S295, S298, S299, S334, S1159, S1160, S1325, S1328
\oval I450, I453, I812, I829
\over x469, E149, E341
\overbrace x536
\overfullrule b341, O69
\overleftarrow x533
\overrightarrow x530
\owns x431, x432

P

\P o310
package/after *681*
package/after/... . . . . *681*
package/before *681*
package/before/... . . . . *681*
\PackageError c72, c101, c110, j84, o1507, A736, A786, A830
\PackageInfo j84, A738, A755, A760, A776, A777, A837, A1122, R50
\PackageWarning j84, A737, A787, A1051, R41, R59, R397
\PackageWarningNoLine . . j84, o969, S1959

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\pagebreak *173*
\pagebreak l6, l7, l22, l24
\pagegoal S1880, S1887
\pagenumbering *454*
\pagenumbering B5
\pageref C10
\pageshrink S538, S542, S558
\pagestyle O2
\pagetotal S128
\paperheight S93
\paperwidth S93
\par *506*, a120, b11, b378, b386, b387, b402, b411, b412, b413, b415, b417, b419, d156, f9, f21, k3, k4, k5, r642, D154, D311, D318, D410, D431, F63, F110, F127, F129, F135, F161, F164, G269, G290, G336, H195, H385, K41, K90, K205, K227, L15, L24, L249, O62, O63, R316, S166, S193, S257, S1886
\paragraphmark K143
\parallel x401
\parbox *512*
\parbox G215
\parboxrestore G303
\parfillskip . . . b356, r616, r631, r676, D347, D357, D366, D374, D406, D428, F76, G277, G298, K192, K215
\parindent b348, b427, b428, D347, D352, D357, D366, D370, D374, D406, D428, F50, G272, G293, K193, K216
\parsep *498*, F1, F49, F90
\parseunicodedataI d123, d162
\parseunicodedataII d124, d126
\parseunicodedataIII d128, d134
\parseunicodedataIV d130, d142
\parseunicodedataV d146, d149
\parshape F54
\parskip *498*, b349, D322, D404, D406, D426, D428, E496, F49, F73, F88, F90, F117, F153, F172, F223, G272, G293, H79, S1159, S1327
\partial x315
\partopsep *498*, E494, F1, F61
\PassOptionsToClass *635*
\PassOptionsToClass P345
\PassOptionsToPackage *635*
\PassOptionsToPackage P345
\patterns o205
\pdffilesize e52, e106
\pdfhorigin R238
\pdfvariable R237, R242
\pdfvorigin R243
\penalty b407, b408, b409, b410, b411, b412, b416, b418, b420, l34, l37, l46, l281, l291, l316, l320, z118, D412, D415, D433, D436, E37, E194, E390, E400, F190, H56, L195, L199, L201, L217, L221, L223, N17, S136, S176, S195, S198, S1157, S1323
\perp x447
\phantom E75
\Phi x305
\phi x287
\Pi x302
\pi x282
picture (environment) I21
\picture I21
\pkg R445, R449
\pm x390
\pmatrix E160, E168
\pmod E39
\poptabs j206, H142, H161
\poptracing t130, t294
\postdisplaypenalty l12, E437, E449, E475
\pounds o318
\Pr E32
\prec x421
\preceq x424
\predisplaypenalty b327, E436, E448, E474
\pretolerance b314, r618, r633, r678
\prevdepth *185*, b398, b402, b403, l287, l288, l349, l354, l383, l388, E192, L196, L198, L218, L220, S167, S169, S172
\PreviousTotalPages . *703*, *716*, *716*, R322
prg commands:
  \prg_do_nothing: Q129, Q131, R105, R245
  \prg_new_conditional:Npnn g254, g379, g829, g841, g847, g853
  \prg_new_protected_conditional:Npnn g234, g317
  \prg_replicate:nn Q479
  \prg_return_false: g238, g251, g262, g266, g269, g321, g384, g837, g845, g851, g858
  \prg_return_true: g249, g265, g321, g382, g836, g839, g844, g850, g856
\prime x253, x317, E243
\ProcessOptions o1528, t71, A753, A790, P399, P466, P908
\ProcessOptions* P399
\prod x349

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

prop commands:
\prop_const_from_keyval:Nn g281, g283, g284, g285
\prop_gclear:N g293, g397
\prop_get:NnN g486
\prop_get:NnNTF g197, g295
\prop_gput:Nnn g199, g202, g433, g828
\prop_gremove:Nn g306
\prop_if_empty:NTF g420, g639
\prop_if_empty_p:N g834
\prop_if_exist:NTF g849
\prop_if_in:NnTF g175, g240, g243, g246, g264
\prop_map_break: g463, g708
\prop_map_inline:Nn .. g404, g451, g458, g460, g608, g642, g703, g705
\prop_new:N g31, g32, g33, g60, g325, g326
\prop_put:Nnn g578, g589
\prop_set_eq:NN g429
\prop_show:N *145*
\propto x398
\protect *469*, *706*, *707*, *708*, *708*, *708*, f102, f220, f234, f243, f248, f251, f252, f254, f255, f260, f261, f266, f269, f270, f295, f333, f361, f533, f553, j197, j199, j200, j206, j212, j219, j227, j230, j236, n212, o26, o32, o51, o55, o209, o217, v507, v1055, w488, z143, C12, D182, D192, D227, D230, D245, D255, H264, K12, K72, K82, K164, K171, K177, L17, N5, R44, R56, R87, R91, R94, S596, S655, U256
\protected *86*, *228*, *309*, *362*, f7, f443, l56, l462, o308, o309, p160, s550, s554, s557, s560, s563, s566, s569, s572, v897, w423, D115, D310, R417, R418, R419, R443, R444
\providecommand f178, g973, g976, o6, o964, A723, A724, S1970
\ProvideCommandCopy *88*
provides commands:
provides_module d290
\provides_module *41*
\ProvidesClass *635*
\ProvidesClass P325
\ProvidesExplPackage Q468
\ProvidesFile a89, x665, x667, x668, x669, P334
\ProvidesPackage *635*
\ProvidesPackage *644*, *645*, t13, A729, A758, P248, P327, P329, P1509, Q497, R412, R440
\ProvideTextCommand o3, o60
\ProvideTextCommandDefault o57
\Psi x306
\psi x289
\pushtabs .. j206, H138, H139, H158, H160
\pushtracing t115, t275
\put *701*, *704*, *713*, I56, I58, I70, I72, I325, I326, I327, I328, I336, I338, I351, I352, I353, I354, I361, I364, I384, I385, I386, I387, I393, I395, I407, I408, I409, I410, I415, I420, I729, I785, I813, I830

Q

\qbezier *552*
\qbezier I682, I814, I831
\qbeziermax I681, I707, I708, I768
\qquad l473
\quad l473, E155, E157, E177, K111
quark commands:
\quark_if_recursion_tail_stop:n g909
\q_recursion_stop g915
\q_recursion_tail g914, g915
quark internal commands:
\s__file_stop Q102, Q104, Q107, Q108, Q110, Q122
\s__hook_mark g40, g106, g109, g112, g116, g119, g120, g221, g255, g274, g275, g279, g776, g789, g794, g802
\quotedblbase o552, o777, o1142
\quotesinglbase o553, o1139

R

\r b376, b377, o236, o388, o428, o467, o604, o631, o641, o667, o750, o789, o1211, o1229, o1255, o1377, o1378, A191, A209
\radical v921, v924, v954
\raggedbottom O53
\raggedleft D353, D371, D382, D389
\raggedright D348, D367, D381, D387
\raise o327, o359, o427, o430, o726, o791, o878, o1219, w501, x465, x513, x515, E73, G412, G421, I61, I73, I105, I117, I195, I305, I452, I495, I526, I551, I619, I636, I637, I740, I796
\raisebox *513*
\raisebox o1188, G389
\rangle x592
\rbrace o309, x596
\rbrack b374
\rceil x600
\Re x313
\read *210*

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\ReadOnlyShipoutCounter *703*
\ReadonlyShipoutCounter *703*, *703*, R270, R353
\Ref *456*
\Ref C47, C52, C59, C63, C70
\ref *509*, C10, C47, C59
\refstepcounter *258*
\refstepcounter *491*, C39, C40, C52, C54, C63, C65, E337, E476, F202, J27, K59, L9
\registernumber *40*
registernumber d381
\relax *81*, *93*, *126*, *146*, *303*, *469*, *658*, *691*, *718*
\Relbar x476, x484, x486, x492
\relbar x473, x488, x490
\relpenalty b322
remove commands:
  remove_from_callback d746
\remove_from_callback *41*
\RemoveFromHook *102*, *102*, g905, g975
\removelastskip .. b414, b416, b418, b420
\renewcommand *71*
\renewcommand . *88*, *369*, *375*, *395*, f124, x66, x68, x70, x71, x73, x75, x77, x78, x84, x86, x88, x89, x103, x104, x105, x113, x114, E424, E444, E465
\RenewCommandCopy *88*, f461, f463
\renewenvironment *72*
\renewenvironment f152, E473, E485
\repeat a81, a83, b390, d154, d164, r659, H382, P957, P1020, P1088, P1151, P1202, P1240, U267, U278, U288, U299, U329, U355, U365
\requestedLaTeXdate P1303, P1336, P1356, P1442
\requestedpatchdate P1366, P1443
\RequirePackage *635*
\RequirePackage .. *680*, *703*, d24, e125, P499, P506, P533, P542, P904, S1967
\RequirePackageWithOptions *635*
\RequirePackageWithOptions P525
reserved@a commands:
  \reserved@a: n234, n305, n352, P934, P1065
reserved@b commands:
  \reserved@b: P216, P233
reserved@c commands:
  \reserved@c: n608
\reservedb *269*
\restorecr l480
\reversemarginpar L363
\rfloor x604
\rgroup x608
\rhd w546
\rho x283
\rhook x479, x480
\right ... x625, x627, x629, x631, x636, x637, x638, x639, E155, E160, E184
\Rightarrow x411, x486, x498
\rightarrow x438, x440, x444, x478, x488, x496, x549
\rightarrowfill x531, x547
\rightharpoondown x455
\rightharpoonup x454, x466
\righthyphenmin T11
\rightleftharpoons x464
\rightline G428
\rightmargin *498*, F9, F40, F51
\rightmark O48
\rightskip .. b428, r630, D345, D349, D355, D365, D368, D373, D405, D427, F75, G276, G297, K192, K215
\rlap o427, o430, o791, E414, E425, G432, H81
\rmdefault *256*, *373*, *373*, *377*, *378*, w6, w143, w191, w239, w271, w303, w332, x50, x120, A7, A48
\rmfamily *119*, *368*, *372*, *381*, w4, w5, w237, w301, w302, w330, w331, z15
rmfamily w252
\rmoustache x565
\rmsubstdefault x18, x30, A28, A39, A106
\Roman *258*
\Roman *703*, p104
\roman *258*
\roman p103
\romannumeral *470*, p109, p110, D181, D198, D244, F43, F234, F245
\root E66, E343
\rootbox E66
\rq b372
\rule *513*
\rule G360, G363, L448

S

\S o311
\samepage *173*
\samepage l11, l28
\savebox *512*
\savebox G97
\savecatcodetable d117, d168, d170
\sb E199
\sbox *512*
\sbox b422, m4, o495, o511, F205, G103, G110, G114, G119, G124
scan commands:
  \scan_new:N g40

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\scan_stop: g230, g235, R249, R251, R252
\scdefault s562, w27, x94
\scriptfont t292
\scriptscriptfont t293
\scriptscriptstyle E65, E68
\scriptspace b347
\scriptstyle x337, E64
\scshape .. o300, s560, s561, w25, w26, z23
\searrow x406
\sec E20
\secdef K142
\sectionmark K143
\selectfont *119*, *377*, *378*, *378*, *387*, m7, o302, o329, o360, o446, o803, o1187, o1221, o1529, r285, s451, s456, s461, s552, s556, s559, s562, s565, s568, s571, s574, t112, w6, w9, w12, w15, w18, w21, w24, w27, w30, w216, w235, w241, w246, w251, w287, w298, w305, w310, w315, w326, w329, w332, w335, w338, w414, w491, w514, A36, A115, A125, A552, A584, A825, A842, L379, L400
seq commands:
\seq_clear:N g448
\seq_clear_new:N g456
\seq_get:NNTF g926
\seq_gpop:NN g925, g935
\seq_gpop:NNTF Q71
\seq_gpush:Nn g916, g919, Q63
\seq_gput_right:Nn g48, g910
\seq_map_inline:Nn .. g398, g472, g490
\seq_new:N g28, g35, g441, Q60
\seq_put_right:Nn ... g454, g546, g553
\seq_use:Nnnn g606, g611
\seriesdefault . *373*, *389*, o1530, v271, w155, w157, w510, w524, w586, x120
\setattribute *39*
\setattribute d82, d231
\setcounter *258*
\setcounter n383, p2, p37, F225, S2290, S2293, S2296, S2300
\setlength *266*
\setlength .. *178*, *187*, l83, l243, l442, q4, E492, E497, E498, E499, G43, G185, G246, G249, G322, G379, G380, G381, G410, G411, G418, G419, G420, H176, H384, S2306, S2307, S2308, S2311, S2312, S2316, S2317, S2318, S2322, S2323, S2324
\SetMathAlphabet r11, u140, u141, v512, x155, x156
\setminus x393
\setrangecatcode .. d96, d104, d113, d114
\SetSymbolFont ... v367, x145, x146, x147
\settodepth *266*
\settodepth q17
\settoheight *266*
\settoheight q17
\settowidth *266*
\settowidth q17
\sfcode . b364, b365, b366, b367, b451, l416, n47, n116, n174, U183, U448
\sfdefault w9, w147, w192, w244, w272, w308, w335, x50
\sffamily *119*, *368*, *378*, *381*, w7, w8, w242, w306, w307, w333, w334, z16
sffamily w252
\sfsubstdefault x19, x31, A30, A108
\shapedefault *306*, *373*, *383*, *396*, o1530, s552, v272, w511, w525, x120
\sharp x330
\shipout *118*, *700*, *700*, *702*, *706*, *706*, *706*, *707*, *708*, *718*, R4, R31, R345, R348, S602, S660
shipout commands:
\l_shipout_box_dp_dim *700*, R118, R121, R123, R154, R207, R432
\l_shipout_box_ht_dim *700*, R117, R120, R123, R153, R173, R206, R431
\l_shipout_box_ht_plus_dp_dim *700*, R120, R123, R139, R190, R201, R203
\l_shipout_box_wd_dim *700*, R119, R123, R165, R217, R433
\shipout_debug_off: .. *703*, R7, R329
\shipout_debug_on: ... *703*, R7, R328
\shipout_discard: R267, R326
\shipout_discard_box: *702*
\g_shipout_readonly_int *703*, R64, R66, R78, R81, R270, R279, R283, R287
\g_shipout_totalpage_int *703*
\g_shipout_totalpages_int *703*, *707*, R48, R272
shipout internal commands:
__shipout_add_background_box:n R104, R130, R262, R334
__shipout_add_background_picture:n R74, R261, R338
__shipout_add_firstpage_material:Nn R106, R113, R327, R332
__shipout_add_foreground_box:n R82, R181, R265, R336
__shipout_add_foreground_picture:n R71, R264, R340

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

__shipout_debug:n *705*, R7, R65, R80
\g__shipout_debug_bool R6, R10, R15, R21
__shipout_debug_gset: R7
\g__shipout_discard_bool R43, R49, R51, R127, R268
__shipout_excuse_extra_page: R299, R307
__shipout_execute: . . *706*, R25, R31
__shipout_execute_cont: . . R37, R39
__shipout_execute_firstpage_-hook: *708*, R77, R102
__shipout_execute_test_level: R28, R33
__shipout_get_box_size:N *710*, R46, R69, R116, R131
\l__shipout_group_level_tl R26, R32, R35
\c__shipout_horigin_tl . . R234, R249
__shipout_init_page_origins: R234, R248
\g__shipout_lastpage_handled_-bool R85, R112, R289
__shipout_picture_overlay:n R247, R262, R265
\l__shipout_saved_badness_tl R128, R134, R142, R155, R160, R168, R177, R185, R193, R205, R212, R222, R230
__shipout_saved_protect: R44, R91, R94
\l__shipout_tmp_box R128, R141, R143, R144, R145, R149, R167, R169, R170, R171, R174, R192, R194, R195, R196, R202, R221, R223, R224, R225, R227, R253, R255, R256, R257
\c__shipout_vorigin_tl . . R234, R251
shipout/background *701*, R95
shipout/before *701*, R95
shipout/firstpage *701*, R95
shipout/foreground *701*, R95
shipout/lastpage *701*, R95
\ShipoutBox *700*, *701*, *702*, R23, R352, R415, R464
\ShipoutBoxDepth R372, R432
\ShipoutBoxHeight *720*, R371, R430
\ShipoutBoxWidth R373, R430
\ShipoutoBoxDepth R430
\shortstack . . . I132, I147, I152, I815, I832
\show *89*, *91*, *92*, f505, f575, f576
show commands:
\show_hook:n *142*
\showboxbreadth b336, b454, b507, b524, b540
\showboxdepth b337, b454, b506, b523, b541, r618, r662, r679
\ShowCommand *89*, *91*, f498
\ShowCommmand *86*
\ShowHook *108*, *111*, *113*, g940, g980
\showhyphens r607
\showoutput b453
\showoverfull b452, b455, b489, b497
\showtokens *93*, f611
\Sigma x303
\sigma x284
\sim x445, x457
\simeq x446
\sin E9
\sinh E11
\skew x544
\skip b28, b53, b92, b208, b250, b295, d33, G339, L367, S316, S490
skip commands:
\skip_zero:N R146, R147, R148, R197, R198, R199
\skipdef b45, b53, b92, d221
\skipzero d221
\slash b407, f785, f806
\sldefault s559, w24, x94
\sloppy G281, G300, O57, O62
sloppypar (environment) O62
\sloppypar O62
\slshape o437, o794, s557, s558, w22, w23, z22, A543
\small *102*
\smallbreak b415, f786, f807
\smallint x357
\smallskip b416, l400
\smallskipamount b415, l400, l403
\smash . x473, x547, x548, x551, x552, E126
\smile x450
\sp E199
\space b380
\spacefactor b405, b406, l129, l138, l157, l171, l183, l197, l211, l416, l429, l434, o70, o73, L463, L465
\spaceskip A6, A47
\spadesuit x334
\span H383
\special *118*, *701*
\splitfirstmark S2211
\splitmaxdepth b343, L441, S2205
\splittopskip b355, L440
\sqcap x376
\sqcup x377
\sqrt E342

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\sqrtsign x529, E71, E342
\sqsubset w542
\sqsubseteq x399
\sqsupset w543
\sqsupseteq x400
\SS o304, o531, o1132, U483
\ss . . o254, o406, o554, o778, o1107, U483
\sscdefault s459, s574, s603
\sscshape s459, s572, s573, s602, z31
\stackrel E340
\star x397
\stepcounter *258*
\stepcounter p17, p27, r572, v48, C41, C54, C65, E350, E409, E486, L428, L452, S646, S705
\stop D154
\storedpar d156, d161
str commands:
  \str_if_eq:nn *134*
  \str_if_eq:NNTF Q482
  \str_if_eq:nnTF e133, g104, g257, g292, g599, g660, g737, g796, g864, g881, Q128, Q476, Q486
  \str_if_eq_p:nn g261, g483
  \str_map_function:NN e137
  \str_set:Nn Q335, Q336
str internal commands:
  __str_if_eq:nn *120*, g23
\strcmp P250, P266
\stretch l456
\string *688*, *691*
\strut . . b423, f787, f808, E178, E179, H29
\strutbox b423, t143, G360, H186, H187, L441, L448
\subparagraphmark K143
\subsectionmark K143
\subset x426
\subseteq x428
\subsubsectionmark K143
\succ x420
\succeq x423
\sum x350
\sup E24
\suppressfloats S1972
\supset x425
\supseteq x427
\surd x336
\swarrow x408
\swdefault s454, s571, s601
\swshape s454, s569, s570, s600, z30
\symbol o162, w469
\symletters A8, A14, A49, A1048
\symoperators x643

T

\T j23, o335, o337, o339, o341, o343, o345, o347, o349, o351, o374, P1283, P1287, P1288
\t o282, o735, o843, A177, A178, A192, A194, A195, A207, A210, A212, A598, A768, A1035, A1037
tabbing (environment) H71
\tabbing H71
\tabbingsep H130, H132, H166
\tabcolsep H259, H338
\tableofcontents *464*
\tabskip b434, E195, E196, E355, E358, E361, E363, E490, E503, E506, E508, H167, H192
tabular (environment) H174
\tabular H174
\tabular* H175
\tabularnewline H194, H207
\tan E15
\tanh E17
\tau x285
\tencirc y10, I125, I678
\tencircw y10, I128
\tenln y9, I124, I126, I677, I679
\tenlnw y9, I127, I129
\TeX m1, m12
TEX and LATEX 2ε commands:
  \...-h@@k *657*
  \@ . . . a65, d20, d859, f728, f729, j19, l425, m2, P45, P60, P71, P100, U415
  \@...hook *123*, *124*
  \@@ a331, a332, i15, i19, i20, i21, i22, i24, i27, i28, i30, i31, n606, n622, t464, t466, t467, H238, H239, H240, H250, S10, S11
  \@@defaultsubs r556
  \@@enc@update o183, r259, r263
  \@@end *210*, a69, a222, e86, f23, n583, n584, D31, D100, D154, T18, U502, U523
  \@@endpbox H193, H236, H386
  \@@eqncr E367, E389, E399, E404, E509
  \@@fileswith@pti@ns P443, P462, P840
  \@@hyph f24, f745
  \@@hyphenation o205
  \@@if@newlist . S599, S644, S657, S703
  \@@ifdefinable f132, o17
  \@@input a68, f22, n481, n494, n534, D25, D80, P1430, Q168, Q181, Q189
  \@@italiccorr f25, z113, z117
  \@@line G428
  \@@math@bgroup z131, z138
  \@@math@egroup z128

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@@par f21, k4, D154, D407, D412, D415, D429, D433, D436, F82, F85, G247, G269, G290, H199, K67, K118, S257
\@@patterns o205
\@@protect f254, f260, f269
\@@startpbox H193, H236, H386
\@@sverb *478*, D486
\@@underline G384, G387, G388
\@@unprocessedoptions P756, P817, P894
\@@warning j166
\@Alph *96*, p106, p122
\@DeclareEncodingSubset A66, A68, A69, A70, A71, A74, A81
\@DeclareMathDelimiter . . . v784, v803
\@DeclareMathSizes . . r206, r207, r209
\@Esphack l190, L201, L223, L241
\@IncludeInRele@se c79, c80
\@IncludeInRelease c77, c78, c79
\@M b21, b408, b409, f37, f39, l11, l12, l13, l14, l15, l16, l17, l18, l119, r625, r632, t393, t406, E377, F194, H56, K67, K100, K118, K130, K194, K217, S176, S195, S198, S258
\@MM *155*, b21, L441, S299
\@Mi h3, S136
\@Mii h3, L53, L122, L194, L216, L241, L311, S295, S1159, S1326
\@Miii h3, L55, L124, L313, S298
\@Miv h3, L195, L201, L217, L223, S272
\@Roman p104, p110
\@TeXversion *2*, a326, j28
\@abspage@last *714*, *715*, *715*, *716*, R67, R78, R275, R277, R279, R287, R324, R325
\@acci w588, G271, G292
\@accii w588, G271, G292
\@acciii w588, G271, G292
\@acol H168, H178, H260, H261, H273, H274, H277, H294, H309, H317, H327
\@acolampacol H258, H275, H277, H284, H292, H326, H329
\@activechar@info S575
\@addamp H251, H260, H261, H276, H290, H327, H328
\@addfield H43, H53, H86, H93, H125, H140, H142
\@addmarginpar S331, S1811
\@addtobot S975, S1062, S1129, S1181, S1290, S1349
\@addtocurcol S328, S1066, S1965
\@addtodblcol S854, S1562
\@addtofilelist a101, a103, n66, n134, n189, n481, n603, w562, w565, w572, w575, w582, w585, Q165, Q181, Q189, U212, U215, U543
\@addtonextcol . . . S853, S1386, S1966
\@addtopreamble H311, H324, H330, H331, H332, H334, H346
\@addtoreset . . p16, p39, p44, p79, p82
\@addtotoporbot S1012, S1175, S1343, S1435, S1524
\@afterheading K92, K125
\@afterindentfalse K45
\@afterindenttrue K43, K124, K193, K216
\@alph *96*, p105, p118, L375
\@ampacol H258, H275, H286, H329
\@arabic p43, p76, p87, p102, p108, L373
\@argarraycr H203, H204
\@argdef f80
\@argrsbox G408
\@argtabularcr H210, H211
\@array H181, H182
\@arrayacol H168, H258
\@arrayclassiv H169, H331
\@arrayclassz H168, H275
\@arraycr H170, H201, H203
\@arrayparboxrestore G261, G303, H384
\@arrayrule H309, H311, H315, H317, H319, H346
\@arstrut H192, H237, H343
\@arstrutbox . H185, H218, H343, H385
\@author K8, K32
\@auxout . . n218, n224, n268, n286, n309, n338, n356, n371, C33, K181, N7, N8, N19, N29, N37, N43, R286
\@backslashchar f219, f469, f582, f594, f598, f599, f604, j185, j187, x266, P1021, P1152, P1241
\@badcrerr j227
\@badend j198, D283
\@badlinearg j217, I165, I177, I188, I189, I193, I242, I247, I257, I262, I275
\@badmath . . j201, E262, E264, E269, E272, E281, E293, E298, E307, E320, E325, E433, E445, E461, E470
\@badpoptabs j205, H85, H151
\@badrequireerror P379, P902
\@badtab j208, H22, H87, H108, H114, H121, H148
\@begin@tempboxa G27, G42, G184, G247, G409, G417

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@begindocumenthook *115*, *123*, *124*, n63, n128, n131, n183, n186, N33, P851, P872
\@begindvi *705*, S623, S682, S710
\@begindvibox *701*, R362, R363, S86, S711
\@beginparpenalty *498*, l14, E436, E448, E474, F23, F170
\@begintheorem J30, J35
\@bezier I683, I684
\@bibitem N3, N8
\@biblabel N4, N54
\@bitor S15, S881, S901, S937, S960, S1027, S1111, S1121, S1269, S1280, S1422, S1509, S1627, S1752
\@botlist S65, S384, S386, S431, S433, S717, S738, S747, S748, S989, S992, S1027, S1121, S1280, S1922, S1949
\@botnum L274, S109, S986, S987, S992, S996, S1458, S1463, S1551, S1558, S1914, S1941, S1983
\@botroom L275, S110, S989, S992, S1915, S1942
\@boxfpsbit S2031, S2033, S2038
\@break@tfor i31, n433, n471, z98
\@bsphack l36, l125, l340, l356, l374, l390, C32, L52, L121, L310, M6, M18, M23, M35, N39, S1882
\@caption L12, L14
\@captype L5, L9, L12, L40, L88, L109, L157, S1995
\@car f53, m14, o89, o110
\@carcube f55, f135, f585
\@cclv b16, S300, S304, S382, S383, S412, S429, S430, S459, S483, S487, S488, U53
\@cclvi b21, b57, b82, b93, b95, b99, b159, b173, d30, d58, P1019, P1150, P1239
\@cdr f53, f671, f672, n252
\@centercr D306, D344, D349, D354, D364, D368, D372
\@centering E347, E348, E355, E358, E361, E502, E506
\@cflb S714
\@cflt S714
\@changed@cmd o3, o63, o223, r131, r267, U374
\@changed@x o3, o211, o219
\@changed@x@mouth o211, o219
\@charlb n377, n385
\@charrb n379, n385
\@chclass H271, H272, H335, H348, H353
\@check@IncludeInRelease c105, c107, c109
\@check@c f189, f191
\@check@eq f195, f196, f200
\@checkend D15, D72, D202, D209, D261, D276, D282
\@chnum H279, H298, H335, H350, H351, H352
\@circ I622, I640, I649, I654, I657
\@circle I605, I606
\@circlefnt I125, I128, I447, I491, I520, I545, I615, I631, I663, I678
\@cite N16, N52
\@cite@ofmt N24, N53
\@citea N15, N17
\@citeb N18, N19, N20, N23, N24, N42, N43, N44, N45
\@citex N13, N14
\@classi H271, H307
\@classii H271, H321
\@classiii H271, H326
\@classiv H169, H180, H272
\@classoptionslist P9, P408, P419, P598, P599, P626, P627, P1511
\@classv H272, H332
\@classz H168, H179, H271
\@cline H367
\@clnht I195, I196, I204, I206, I208, I218, I225, I273, I672
\@clnwd I197, I203, I207, I209, I210, I672
\@cls@pkg P257, P258, P271, P272, P669, P715, P753, P796, P826, P878, P887, P889, P906, P1364, P1439, P1461, P1491
\@clsextension *657*, *688*, P30, P129, P140, P203, P287, P303, P315, P375, P392, P408, P418, P500, P515, P523, P597, P625, P719, P746, P773, P830, P843, P879
\@clubpenalty n9, n27, n96, n153, F128, F196, K106, K135
\@colht n24, n93, n150, L273, L275, L278, L284, L285, L298, L299, S114, S231, S242, S251, S252, S387, S399, S434, S447, S474, S505, S535, S541, S545, S555, S560, S645, S704, S777, S815, S859, S884, S903, S943, S965, S1642, S1768, S2096, U88
\@colnum . L276, S111, S995, S1040, S1109, S1110, S1138, S1146, S1267, S1268, S1300, S1312, S1420, S1421, S1458, S1463, S1507, S1508, S1550, S1557, S1910, S1937, S1976, S2151

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@colroom . . . n25, n94, n151, S115, S252, S273, S274, S285, S288, S387, S434, S777, S994, S1039, S1105, S1108, S1137, S1262, S1266, S1299, S1416, S1419, S1502, S1506, S1911, S1938, S2106, S2111, S2156, U87
\@combinedblfloats S750, S2230, S2269
\@combinefloats S501, S714
\@comdblflelt S750
\@comflelt S720, S736, S750
\@cons b196, b213, f52, p44, L193, L215, L239, L355, S237, S888, S907, S923, S947, S949, S969, S971, S1141, S1209, S1305, S1378, S1451, S1541, S1644, S1667, S1770, S1795, S1812, S1813, S2157
\@contfield H50, H141, H153
\@copy@... . . . . *90*
\@copy@DeclareRobustCommand f483, f520, f541, f616, f619
\@copy@newcommand *83*, *91*, f306, f484, f520, f562, f591, f616, f622
\@ctrerr . j194, p121, p125, p139, p147
\@curfield H16, H41, H47, H51, H52, H54, H130, H131
\@curline H16, H27, H39, H44, H53, H54, H55, H90, H91, H103, H128, H129
\@curr@enc o154, o156
\@curr@file *656*, *656*, *691*, *691*, *692*, *693*, n228, n229, n238, n240, n264, n272, n390, n409, n501, n516, P700, P964, P969, P975, P998, P1007, P1033, P1095, P1100, P1106, P1129, P1138, P1164, Q258, Q310, Q312
\@curr@file@reqd *691*, *693*, Q258, Q312, Q316
\@currbox b275, b276, b277, L60, L91, L95, L129, L160, L164, L193, L214, L215, L239, L257, L259, L261, L319, L322, L327, L331, S213, S214, S225, S226, S228, S229, S237, S311, S312, S853, S854, S1102, S1104, S1112, S1135, S1139, S1141, S1156, S1197, S1209, S1257, S1260, S1297, S1302, S1305, S1322, S1367, S1378, S1410, S1426, S1440, S1451, S1493, S1530, S1541, S1581, S1585, S1596, S1602, S1604, S1608, S1613, S1622, S1631, S1637, S1644, S1667, S1702, S1706, S1718, S1725, S1727, S1731, S1737, S1747, S1762, S1770, S1795, S1813, S1822, S2001, S2002, S2031, S2061, S2066, S2112, S2115, S2127, S2135, S2152, S2157
\@currdir *9*, a108, a130, a132, a138, a140, a146, a148, a153, a155, a165, a178, a243, a256, a269, P947, P969, P998, P1078, P1100, P1129, P1212
\@current@cmd o25, r271
\@currentcounter . C39, C40, C42, C53
\@currentlabel *491*, C34, C43, C55, C66, C74, E351, E487, G356, L443
\@currenvir j199, D3, D165, D220, D237, D270, D283, F112, G130, P1009, P1021, P1029, P1033, P1039, P1140, P1152, P1160, P1164, P1170, P1229, P1241, P1249, P1253, P1259
\@currenvline j199, D166, D221, D238, D271, D284, G131
\@currext *656*, P29, P44, P59, P70, P99, P283, P286, P287, P302, P303, P314, P315, P392, P401, P408, P418, P445, P510, P519, P531, P665, P666, P667, P672, P678, P685, P690, P692, P697, P703, P711, P717, P719, P726, P729, P734, P737, P739, P740, P742, P746, P755, P757, P758, P763, P766, P769, P773, P779, P788, P793, P794, P799, P805, P809, P811, P812, P814, P816, P818, P819, P822, P828, P830, P843, P856, P879, P895, P896
\@currlist . L193, L215, L355, S67, S311, S388, S391, S435, S438, S1812
\@currname *104*, *104*, *124*, *125*, *151*, *655*, *655*, *656*, *656*, *656*, c66, c82, c90, g126, g128, n615, P28, P43, P58, P69, P98, P255, P257, P269, P271, P283, P286, P302, P314, P401, P445, P519, P531, P663, P666, P667, P669, P670, P672, P678, P685, P687, P690, P692, P697, P702, P711, P715, P717, P726, P728, P734, P737, P739, P740, P744, P748, P755, P757, P758, P763, P766, P770, P774, P779, P787, P811, P812, P814, P816, P818, P819, P856, P887, P889, P896, P906, P1439, P1461, P1491
\@currnamestack *151*, *639*, *698*, g913, P34, Q473
\@curroptions P401, P409, P435, P454, P896, P897

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@currpath . *639*, *644*, *655*, *655*, *658*, P14, P48, P112, P255, P257, P664, P685, P692, P701, P726, P727, P751
\@currpkg@reqd *656*, *656*, P285, P695, P697, P706, P722, P736, P751, P753
\@currsize w489
\@currtype S119, S878, S879, S880, S881, S898, S899, S900, S901, S1027, S1111, S1121, S1269, S1280, S1422, S1509, S1627, S1752, S2001, S2003, S2004, S2007
\@curtab H11, H26, H86, H87, H88, H94, H95, H98, H102, H103, H107, H146, H147
\@curtabmar H11, H25, H26, H38, H44, H89, H102, H106, H107
\@d@r a161, a162
\@dashbox I324, I325, I326, I327, I328, I331, I337, I339, I349, I351, I352, I353, I354, I358, I362, I365, I382, I384, I385, I386, I387, I390, I394, I396, I405, I407, I408, I409, I410, I413, I417, I422, I674
\@dashcnt I317, I319, I320, I321, I322, I323, I336, I338, I341, I343, I344, I345, I347, I348, I361, I364, I376, I377, I378, I379, I380, I381, I393, I395, I398, I399, I400, I401, I403, I404, I416, I421, I674
\@dashdim . . . I316, I317, I318, I319, I320, I322, I325, I327, I328, I329, I336, I338, I340, I341, I342, I343, I344, I347, I351, I353, I354, I355, I363, I366, I375, I376, I377, I378, I380, I384, I386, I387, I388, I393, I395, I397, I398, I399, I400, I403, I407, I409, I410, I411, I419, I424, I674
\@date K9, K33
\@dbflt L32, L264
\@dblarg f693, K54, K142, L12
\@dbldeferlist L239, S70, S445, S450, S452, S816, S823, S824, S1752, S1795, S1926, S1954
\@dblfloat L31
\@dblfloatplacement n33, n102, n160, L280, S401, S449, S1907, S1934, S2235, S2275
\@dblflset L26
\@dblfpbot L290, L304, S2319
\@dblfpsep L289, L303, S2319
\@dblfptop L288, L302, S2319
\@dbltoplist S69, S232, S235, S237, S397, S398, S445, S446, S755, S759, S761, S762, S1639, S1644, S1764, S1770, S1925, S1952
\@dbltopnum L283, L297, S107, S127, S238, S240, S766, S1578, S1579, S1643, S1646, S1674, S1679, S1699, S1700, S1769, S1773, S1802, S1807, S1918, S1945
\@dbltoproom L284, L286, L298, L300, S108, S1581, S1584, S1585, S1594, S1595, S1598, S1601, S1604, S1608, S1612, S1616, S1621, S1641, S1702, S1705, S1706, S1715, S1716, S1717, S1720, S1724, S1727, S1731, S1736, S1740, S1745, S1746, S1767, S1919, S1946
\@dec@text@cmd o3
\@declarecommandcopylisthook *88*, *89*, *89*, f480, f482, f494
\@declaredoptions P8, P382, P405, P421, P440, P459, P849
\@declareoption . . . P380, P381, P389
\@defaultfamilyhook w261, w513, w529
\@defaultsubs r510, r544, r556, D47, D87
\@defaultunits r214, r218, r219, r220, r235, r299, t133, t135, I13
\@defaultunitsset G53, G64, I8, I29, I30, I32, I34, I60, I63, I84, I85, I107, I108, I164, I241, I316, I318, I332, I340, I342, I357, I480, I481, I612, I648, I690, I691, I693, I694, I697, I698, I700, I701, I712, I713, I715, I716, I718, I719, I721, I722
\@defdefault@ds . . . P380, P385, P390
\@deferlist S68, S384, S393, S394, S397, S402, S404, S410, S431, S440, S442, S778, S786, S787, S798, S803, S804, S1111, S1209, S1269, S1378, S1422, S1451, S1509, S1541, S1627, S1667, S1924, S1951
\@definecounter p12, p36, E336, F227, F228, F229, F230, J8, J16, L372, L374
\@depth f26, t145, x558, x559, x561, x562, G383, G427, H187, H219, S1851, I227, I300, I303, I324, I333, I383, I391, I727, I783
\@dir a160, a163, a165, a167, a168
\@disable@packageload@do P690, Q413
\@dischyph f730, f764, G270, G291
\@doclearpage S296, S371
\@documentclasshook . P3, P602, P630
\@doendpe D203, D210, D262, D277, F123
\@dofilelist . . . n612, n628, D37, D82
\@donoparitem F144, F158

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@dot I605, I643
\@dotsep K200, K223
\@dottedtocline . . K185, K211, K212
\@downline I297, I301, I306
\@downvector I268, I306
\@eha f280, f319, f347, j170, j188, j190, j192, j200, j202, j232, n225, n269, n287, o52, o84, r28, r58, r102, r144, r187, r253, r310, t106, v25, v70, v99, v161, v192, v224, v325, v346, v378, v419, v464, v469, v524, v642, v646, v650, v685, v689, v693, v750, v760, v845, v850, v853, v885, v888, v961, v964, v967, v1034, v1040, z146, A24, A102, A824, A833, D164, D219, D236, D269, N47, S1876, S1892
\@ehb j170, j195, j220, j222, j224, S234, S390, S437
\@ehc f128, f155, f469, j170, j227, j230, j236, j238, D495, D510, D525, D538, E408, F220, K31, P1491
\@ehd e69, j170, j197, j204, j207, j209, j215, v118, H100, H109, L6, P564, P753
\@elt *371*, *372*, *381*, f52, n378, o1532, o1534, p20, p35, p53, p56, w76, w87, w89, w90, w115, w355, w357, w358, w368, w555, x17, x25, S8, S11, S15, S27, S30, S31, S32, S33, S38, S39, S40, S41, S42, S43, S44, S45, S47, S51, S57, S58, S59, S60, S498, S720, S731, S736, S746, S758, S760, S788, S805, S825, S844, S857, S864, S915, S918, S927, S1898
\@empty *81*, *269*, *305*, *369*, *375*, *389*, *395*, *640*, i14
\@emptycol S198, S245, S248, S277, S281
\@end@check@IncludeInRelease c106, c108
\@end@tempboxa G36, G45, G189, G260, G415, G425
\@enddocument@kernel@warnings D38, D42, D104
\@enddocumenthook *124*, D71, P851, P873
\@endfloatbox . L190, L211, L236, L248
\@endparenv *506*, F120, F123
\@endparpenalty *498*, l15, E437, E449, E475, F23, F124
\@endpbox H193, H236, H266, H333, H384, H387
\@endpefalse D170, D224, D241, D274, F129, F131, F135, F136, F138, G133
\@endpeltrue F138
\@endpetrue F124, F126, F134
\@endpreamblehook *197*
\@endtheorem J13, J19, J25, J35
\@enlargepage . . . S1861, S1866, S1868
\@ensuredmath E419, E421
\@enumctr F234, F237, F238
\@enumdepth *509*, F226, F232, F233, F234
\@enumspacing *509*
\@eqcnt E344, E405, E410, E489, E504, E505, E507
\@eqncr E356, E374, E411, E412, E491
\@eqnnum E338, E339, E409, E423, E482
\@eqnsel E344, E503
\@eqnswfalse E373
\@eqnswtrue . . E346, E352, E410, E488
\@eqpen E344, E377, E379, E390, E400
\@err@ j37, j41, j44, j52, j64, j68, j71, j79
\@esphack l38, l131, l345, l362, l379, l396, C35, L361, M17, M19, M34, N50, S1884
\@evenfoot O12, O15, S613, S672
\@evenhead O12, O15, S612, S671
\@execute@begin@hook *467*, D167, D171, D174
\@expandtwoargs f217, P207, P254, P407, P421, P469
\@expast H239, H267
\@expl@@@filehook@clear@replacement@flag@@ P353, Q277, Q451
\@expl@@@filehook@drop@extension@@N *692*, Q274, Q275, Q453
\@expl@@@filehook@file@pop@@ P708, Q160, Q457
\@expl@@@filehook@file@pop@assign@@nnnn Q151, Q459
\@expl@@@filehook@file@push@@ P691, Q148, Q455
\@expl@@@filehook@if@file@replaced@@TF *691*, *693*, Q271, Q449
\@expl@@@filehook@if@no@extension@@nTF Q267, Q439, Q441
\@expl@@@filehook@normalize@file@name@@w Q273, Q447
\@expl@@@filehook@resolve@file@subst@@w P351, Q270, Q445
\@expl@@@filehook@set@curr@file@@nNN *655*, *691*, *693*, P350, P657, Q310, Q316, Q443
\@expl@@@hook@curr@name@pop@@ g954, P78
\@expl@@@hook@curr@name@push@@n g954, P40
\@expl@@@initialize@all@@ g954, D175

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@expl@@@shipout@add@background@box@@n R331, R423
\@expl@@@shipout@add@background@picture@@n R331, R427
\@expl@@@shipout@add@firstpage@material@@Nn R331, R420
\@expl@@@shipout@add@foreground@box@@n R331, R425
\@expl@@@shipout@add@foreground@picture@@n R331, R429
\@expl@char@generate@@nn . e138, f709
\@expl@cs@argument@spec@@N e135, e146, f610
\@expl@cs@prefix@spec@@N e134, e145, f607
\@expl@cs@replacement@spec@@N e136, e147, f568, f602, f610
\@expl@cs@to@str@@N . . e129, e132, e141, e143, f477, f534, f554, f556, f557, f570, f582, f594, f598, f599, f604
\@expl@finalise@setup@@ e11, U201, U202
\@expl@pop@filename@@ e7, P77, P81, P87
\@expl@push@filename@@ *639*, e5, P38, P41, P54, P56, P66, P85, P91
\@expl@push@filename@aux@@ e6, P38, P50, P62, P66, P91
\@expl@str@if@eq@@nnTF e133, e144, f441, f443, P254
\@expl@str@map@function@@NN *96*, e137, e148, f706
\@expl@sys@load@backend@@ e3, e4, n19
\@extra@page@added D58, R302
\@failedlist . . . S842, S865, S881, S888, S901, S907, S923, S937, S960
\@fcolmadefalse S833
\@fcolmadetrue S921
\@file-subst@⟨*file*⟩ *689*
\@filef@und *688*, n420, n430, n457, n468, n481, n534, Q156, Q168, Q181, Q189
\@filehook@file@push *656*
\@filehook@set@CurrentFile n315, P693, Q149, Q299
\@filelist n65, n133, n188, n602, n603, n614, w562, w572, w582, U212, U527, U543
\@fileswfalse n201, P943, P944, P1074, P1075
\@fileswith@pti@ns P379, P443, P462, P589, P590, P594, P596, P623, P624, P651, P840
\@fileswith@ptions P584, P585, P587, P591
\@fileswithoptions P500, P507, P515, P582
\@fileswtrue n7, P926, P929, P984, P988, P1057, P1060, P1115, P1119
\@finalstrut . G360, G426, H385, L448
\@firstampfalse . . H254, H277, H294
\@firstamptrue H262
\@firstcolfirstmark S2212, S2213, S2217
\@firstcoltopmark S2210, S2218
\@firstcolumnfalse S2202, S2247
\@firstcolumntrue n30, n99, n157, S98, S207, S2221, S2253
\@firstofone *88*, *101*, d12, d100, d108, d166, e15, e42, e57, f212, f465, f470, f471, f474, f475, n127, n182, n397, o68, o153, t300, v53, v81, v142, v173, v204, v795, D70, E417, H372, L10, N18, N42, P682, P709, Q290, Q305, U257
\@firstoftwo . . . *208*, a87, e16, e24, f212, f442, f444, f466, f537, f587, f645, f655, f665, f692, f716, n431, n469, o133, p157, p162, v799, w360, A90, A745, A796, A812, C19, O16, P136, P175, P187, P214, P232, P1500
\@firsttab H2, H74, H75, H76, H106, H118
\@flcheckspace . . . S989, S1025, S2102
\@flfail S865, S916, S937, S947, S960, S969
\@float L26, L32
\@floatboxreset . . . L101, L170, L174
\@floatpenalty L3, L53, L55, L58, L122, L124, L127, L191, L194, L199, L201, L212, L216, L221, L223, L237, L241, L311, L313, L317, L321, L355
\@floatplacement n33, n102, n160, L271, S149, S209, S253, S477, S1908, S1935
\@flsetnum S986, S1022, S1109, S1267, S1420, S1507, S1578, S1699, S2070
\@flsettextmin S1085, S1237, S1406, S1489, S2086
\@flstop S1972
\@flsucceed S858, S866, S915, S949, S971
\@fltovf j223, L93, L162, L322
\@flupdates S992, S1037, S2148

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@flushglue h17, D345, D349, D355, D365, D368, D373, D406, D428, F76, G277, G298
\@fnsymbol p107, p126
\@font@aliasinfo t493
\@font@info r133, r171, r177, r337, r354, r592, t30, t38, t46, t74, t87, t126, t154, t168, t179, t193, t209, t215, t228, t235, t242, t247, t257, t269, t281, t445, t457, t462, t469, t499, t512, t520, v234, v249, v283, v329, v398, v404, v448, v461, v544, v633, v676, v741, v835, v1002, v1031, A76, A77, A120, U376
\@font@series@contextfalse w352, w388
\@font@series@contexttrue w371, w375, w387
\@font@shape@subst@warning s408, s411, s416, s542, s545, s589
\@font@warning r3, r506, r511, r538, r545, s419, t19, t33, t41, t49, t61, t77, t430, t444, t456, t461, t468, t511, t519, u30, D44, D84, U232
\@fontenc@load@list o1533, w555, x17, x25
\@fontswitch z126, z128
\@footnotemark L430, L436, L454, L460, L461
\@footnotetext G330, L430, L436, L437, L470, L476
\@for i16, n234, n305, n352, n614, N16, N41, P216, P233, P405, P419, P435, P440, P454, P459, P480, P490, P897, P934, P1065
\@forced@seriesfalse s387
\@forced@seriestrue s386
\@forloop i19, i20
\@fornoop i15, i23, i29
\@fortmp i17, i18, i26, P478, P480, P933, P934, P1064, P1065
\@fpbot L290, L304, S863, S2313
\@fpmin L278, L287, L301, S113, S920, S1916, S1943, S2165, S2182
\@fps L41, L42, L44, L47, L64, L110, L111, L113, L116, L133, S1993, S1995, S1998
\@fpsadddefault L45, L48, L114, L117, S1990
\@fpsep L289, L303, S861, S870, S942, S964, S2313
\@fpstype S983, S1004, S1005, S1019, S1050, S1051, S1075, S1077, S1080, S1082, S1133, S1189, S1190, S1225, S1228, S1231, S1234, S1295, S1357, S1358, S1396, S1398, S1401, S1403, S1477, S1480, S1483, S1486, S1575, S1590, S1592, S1610, S1619, S1655, S1656, S1696, S1711, S1713, S1733, S1743, S1782, S1783, S1986, S2002, S2004, S2006, S2009, S2010, S2011, S2013, S2014, S2018, S2019, S2021, S2022, S2056, S2058, S2060, S2072, S2074, S2088, S2090, S2120, S2123, S2134
\@fptop L288, L302, S860, S2313
\@frameb@x G160, G188, G190
\@framebox G167, G174, G178
\@framepicbox G167, G174, G211
\@freelist b196, b213, b275, L60, L129, L319, L320, S29, S34, S48, S56, S213, S499, S732, S747, S761, S866, S1812, S1813
\@getcirc I437, I485, I514, I541, I613, I629
\@getfpsbit S980, S1016, S1572, S1693, S2029
\@getlarrow I266, I274, I276
\@getlinechar I190, I229
\@getpen l34, l37, l46, l117
\@getrarrow I267, I274, I283
\@glossaryfile M21, M22, M31
\@gnewline l95, l101, l108, l111
\@gobble *381*, *656*, d11, d98, e14, e46, e69, f111, f133, f208, f219, f236, f240, f277, f283, f286, f296, f316, f322, f325, f334, f344, f350, f353, f362, f380, f384, f386, f387, f389, f397, f401, f403, i6, i9, j101, j127, j153, j162, l75, l483, n66, n134, n189, n404, n406, n602, o29, r507, r540, t299, u26, v28, v30, v287, v298, v362, v409, v410, v439, v445, v453, v458, v476, v490, v500, v509, v522, v539, v548, v622, v624, v628, v636, v670, v679, v731, v733, v744, v828, v838, v919, v924, v993, v1024, w74, w115, w368, w565, w575, w585, A773, D228, K143, K144, K145, K146, K147, K182, L7, N11, N25, N26, P552, P688, P918, P978, P1007, P1109, P1138, P1222, P1227, P1301, P1445, P1457, S619, S620, S621, S678, S679, S680, S927, S1900, S2166, S2183, U215, U309, U466, U475, U543
\@gobble@AddToHook@args . . g972, g973

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@gobble@IncludeInRelease c86, c94, c103
\@gobble@RemoveFromHook@arg g975, g976
\@gobblecr l481, l482
\@gobblefour f208, v24, v284, v400, v402, v406, v408, v418, v422, v546, v598, P1009, P1140, P1229
\@gobblethree f208, f596, f609
\@gobbletwo . e112, e117, f175, f176, f208, i12, n34, n103, n161, r512, r546, v132, D22, D45, D77, D85, O11, O13, P977, P1108, P1221, U238
\@gtempa f126, f127, f181, f183, f528, f536, n546, n547, n553, n554, n555, n580, n581, n583, n584, n585, H3, H5, H6, H7, H8, P253, P255, P268, P269, P288, P290, P304, P306, P316, P318
\@halfwidth I2, I126, I129, I131, I227, I299, I302, I324, I333, I349, I361, I364, I383, I391, I405, I416, I421, I680, I706, I725, I726, I727, I766, I781, I782, I783
\@halignto ... H170, H174, H177, H191
\@hangfrom K66, K117, K138
\@height b402, f26, l350, l358, l384, l392, o293, o295, t144, x340, x558, x559, x561, x562, G144, G149, G197, G207, G383, G427, H186, H219, H359, H376, S1851, I227, I300, I303, I324, I333, I351, I359, I383, I391, I407, I414, I590, I600, I726, I782
\@highpenalty l118, U3
\@hightab H11, H21, H23, H74, H86, H95, H96, H111, H146, H147
\@hline I167, I179, I226, I265
\@holdpg S122, S300, S302, S303, S308, S309, S310
\@hspace l437, l438, l453
\@hspacer l437, l452
\@hvector I244, I259, I265
\@icentercr D323, D324
\@iden f215
\@if f171, f172, f174
\@if@DeclareRobustCommand *90*, *91*, f483, f507, f518, f519, f523, f614, f615, f618
\@if@newcommand *83*, *91*, *92*, f304, f484, f508, f519, f561, f573, f578
\@if@pti@ns P207, P211, P213, P230, P231, P247
\@if@ptions P202, P203, P206, P208, P667, P794
\@ifatmargin H55, H106
\@ifbothcounters p61, p69, p71, p79, p81, p93
\@ifclasslater P139, P149, P159
\@ifclassloaded P128
\@ifclasswith P202
\@ifdefinable f84, f86, f130, f132, f238, o14, o17, p11, q3, w468, G96, J7, J15, J22
\@iffileonpath n416, n424, n453, n462
\@ifl@aded *655*, P128, P129, P132, P138, P666, P779, P793
\@ifl@t@r o1498, P147, P157, P163, P168, P170, P181, P182, P192, P573
\@ifl@ter o1537, o1538, P139, P140, P162, P711, P822
\@ifl@ter@@ o1537, o1538
\@ifnch e30, e31, f677, f679, f691
\@ifnextchar a98, e26, f673, f678, f692, l93, l482, n494, p13, t365, D322, E342, F143, G9, G11, G18, G20, G26, G47, G102, G103, G109, G110, G117, G121, G166, G167, G173, G174, G179, G212, G220, G228, G235, G239, G308, G312, G316, G367, G372, G395, G402, G407, H57, H181, H203, H210, J3, J5, J28, L27, L264, L324, L428, L451, L468, N3, N13, P260, P274, P568, P583, P588, P927, P930, P1058, P1061, S209, S1974, I23, I132, I143, I453
\@iforloop i21, i22
\@ifpackagelater ... P139, P148, P158
\@ifpackageloaded P128, S1958
\@ifpackagewith P202
\@iframebox G180, G181, G182
\@iframepicbox G212, G213
\@ifstar f73, f692, l59, l71, l330, l437, p67, p77, r206, u121, D311, D318, D549, D558, E376, H56, H202, H209, K52, K142, P380, P402, S1856, I142, I605
\@ifundefin@d@i . f635, f636, f653, f656
\@ifundefin@d@ii f635, f638, f641
\@ifundefined f127, f134, f154, f161, f183, f194, f277, f283, f316, f322, f344, f350, f380, f397, f478, f630, p3, p7, p16, p50, p62, p64, r100, r186, t378, v319, A75, A775, C23, D130, D147, D163, D218, D235,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

D268, J21, N20, N44, O3, O7, P126, P355, P368, P436, P455, Q420
\@ignorefalse D4, D169, D205, D212, D223, D240, D263, D273, D278, L360
\@ignoretrue l200, l213, D4, D7, E335, E338, E370, E512
\@iiiminipage G310, G314, G317, G318, G319
\@iiiparbox G222, G230, G237, G240, G241, G242, G347
\@iiminipage G313, G315
\@iinput *208*, n494, n495
\@iiparbox G236, G238
\@iirsbox G407, G416
\@imakebox G26, G41, G119
\@imakepicbox . G47, G48, G124, G214
\@iminipage G309, G311
\@include . *201*, n229, n272, n288, n292
\@includeinreleasefalse c69, c74, c99, c104, f768
\@includeinreleasetrue c89
\@index M18, M19, M35
\@indexfile M4, M5, M14
\@inlabel *497*
\@inlabelfalse F28, F104, F184, S163, S190
\@inlabeltrue F28, F178
\@inmatherr . . . j233, F112, F142, I605
\@inmathwarn o3
\@inpenc@test U254, U321
\@input *679*, n36, n105, n163, n299, n346, n533, K152, U545
\@input@ . n319, n361, n535, r364, N31
\@input@file@exists@with@hooks Q140
\@inputcheck . *664*, a70, a191, a192, a195, a203, e19, e20, e23, f38, f45, n3, n411, n412, n419, n428, n429, n432, n448, n449, n456, n466, n467, n470, P966, P967, P1003, P1097, P1098, P1134, P1209, P1210, P1217
\@insertfalse S1073, S1223, S1394, S1475, S1570, S1691
\@inserttrue S999, S1044, S1161, S1329, S1649, S1776
\@invalidchar j238
\@iparbox G221, G229, G234
\@irsbox . . . G395, G402, G407, G408
\@isavebox G117, G118
\@isavepicbox G122, G123
\@ishortstack I133, I141
\@istackcr I143, I144
\@itabcr H57, H58
\@item F143, F156
\@itemdepth *509*, F241, F243, F244, F245
\@itemfudge H38, H44, H82
\@itemitem F245, F248
\@itemlabel F44, F96, F143
\@itempenalty *498*, l16, F23, F175
\@itemspacing *509*
\@iwhiledim i7
\@iwhilenum i3
\@iwhilesw i10
\@ixpt r688
\@ixstackcr I142
\@kernel@... . . . . *639*
\@kernel@after@⟨*hookname*⟩ *114*
\@kernel@after@begindocument n60, n77
\@kernel@after@begindocument@before *198*, n18, n77, s613
\@kernel@after@enddocument D14, D40, D102, R276
\@kernel@after@enddocument@afterlastpage D18, D40, D103, R282
\@kernel@after@shipout@lastpage *715*, R83, R100, R293, R298
\@kernel@before@⟨*hookname*⟩ *114*
\@kernel@before@begindocument n58, n77, R322
\@kernel@before@shipout@background R75, R100
\@kernel@currpathstack P47, P49, P80, P105
\@kernel@make@file@csname Q269, Q272, Q299
\@kernel@rename@newcommand . . *83*, *83*, *85*, f285, f302, f339, f367, f372, f385
\@killglue . . . I59, I73, I103, I115, I123
\@kludgeins S319, S320, S321, S323, S376, S377, S423, S424, S502, S518, S519, S525, S526, S527, S536, S552, S556, S566, S1852, S1883
\@labels *497*, F27, F146, F147, F189, F206, F207
\@largefloatcheck L192, L213, L238, L256
\@lastchclass H262, H272, H273, H275, H283, H308, H322, H326, H335, H348, H349
\@latex@error e61, f128, f155, f278, f317, f345, f469, j136, j168, j184, j190, j192, j195, j197, j199, j202, j204, j206, j209, j213, j218, j222, j224, j226, j227, j229, j232, j236, j238, n225, n269, n287, o50, o84, r5, r28, r58, r102, r144, r187, r253, r310, t105, u100, u111, v23, v68, v97, v117, v159, v190, v222, v245, v261, v325, v346, v378, v418, v422,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

v464, v469, v524, v592, v598, v642, v646, v650, v685, v689, v693, v750, v760, v845, v850, v853, v885, v888, v961, v964, v967, v1034, v1040, w46, w57, w450, w533, z143, D164, D219, D236, D269, D495, D510, D525, D537, E408, F219, H100, H109, K31, L6, L83, N47, P511, P546, P559, P668, P752, P795, P886, P903, P911, P916, P938, P993, P1069, P1124, P1460, P1490
\@latex@info f224, f298, f336, f364, f730, j136, o85, A97
\@latex@info@no@line j136, S576
\@latex@warning j136, j166, o55, r33, C14, L260, N22, N45, P1032, P1038, P1163, P1169, P1252, P1258, R107, S1996, I449
\@latex@warning@no@line e79, f202, j136, j167, n21, n90, n147, n600, C8, C26, C27, D52, D59, D92, K32, P256, P270, P574, P712, P823, P968, P974, P997, P1099, P1105, P1128, P1211, P1218, P1285, P1362, S243, S275, S1827, S2062
\@latexbug j225, S333, S1813
\@latexerr j166, S234, S390, S437, S1874, S1891
\@lbibitem N3, N4
\@ldots x504, x506
\@leftcolumn S121, S2203, S2224, S2248, S2257
\@leftmark O16, O50
\@let@token e30, e32, f677, f680, f683, f691, l410, l411, l418, z83, z96, E243, E245, E248
\@lign E195, E197
\@linechar I190, I191, I192, I196, I197, I199, I204, I206, I207, I208, I209, I211, I215, I216, I219, I220, I225, I272, I670
\@linefnt I124, I127, I190, I265, I273, I304, I307, I677
\@linelen I164, I165, I176, I177, I203, I210, I219, I221, I226, I227, I228, I241, I242, I256, I257, I300, I303, I305, I306, I671
\@list *498*
\@listctr *497*, F202, F225, N9
\@listdepth *497*, F23, F35, F38, F43, F99, G331
\@listfiles n64, n132, n187, n606, n621
\@listi *498*
\@listii *498*
\@listvi *498*
\@loadwithoptions P517, P523, P533, P542
\@lowpenalty l117, U3
\@ltab H71, H106
\@m *155*, b21, b362, b364, b365, b398, b399, l288, l429, l434, n47, n116, n174, F80, N17, I213, I217
\@mainaux n5, n39, n40, n108, n109, n166, n167, n218, n299, n338, n346, n371, D21, D76
\@makebox G11, G20, G25
\@makecaption L24
\@makecol S261, S413, S460, S480
\@makefcolumn S393, S394, S402, S404, S440, S442, S450, S452, S2161, S2163, S2179, S2180
\@makefnmark L376, L464
\@makefntext G359, L447
\@makeother . . a76, a97, a126, f695, f696, r377, r378, r379, r380, r381, r382, r383, r384, r385, r386, r387, D417, D438, D531, D546, D556, P342, P343, P1014, P1145, P1234
\@makepicbox G10, G19, G46, I366, I424
\@makespecialcolbox S503, S522
\@marbox L320, L322, L326, L330, L331, L355, S1812, S1822, S1825, S1833, S1835, S1836, S1838, S1839, S1840, S1849
\@marginparreset L339, L346
\@markright O33, O48
\@maxdepth n62, n130, n185, S91, S486, S514, U85
\@maxtab H2, H94
\@medpenalty l118, U3
\@meta@family@list w76, w88, w178, w356
\@midlist . . S66, S499, S500, S1027, S1029, S1141, S1305, S1923, S1950
\@minipagefalse F181, G304, G306, G344, L187, L250, L341
\@minipagerestore G332, G334
\@minipagetrue G305, L186
\@minus f26, S2306, S2307, S2308, S2311, S2312
\@missing@onefilewithoptions P686, P725, P836
\@missingfile@area n507, n548, n561, P727
\@missingfile@base n507, n549, n562, P728
\@missingfile@ext n507, n550, n563, P729

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@missingfileerror *208*, *209*, *209*, *655*, n502, n517, n527, n536, P726, P816
\@mkboth O11, O13
\@mklab F45, F140
\@mkpream H189, H234, H262
\@mparbottom L363, L364, S118, S476, S1823, S1831, S1832, S1833, S1834
\@mpargs G323, G347
\@mparswitchfalse S102
\@mpfn . . G329, L428, L433, L473, L477
\@mpfootins G338, G339, G342, G348, G351, G352
\@mpfootnotetext G330, G350
\@mplistdepth G331, G348
\@multicnt H370, H372, H373, H374, H381, H382, H383, I103, I104, I106, I115, I116, I118, I667, I704, I706, I707, I708, I710, I711, I717, I723, I734, I738, I764, I766, I768, I770, I771, I775, I779, I790, I794
\@multiplelabels n35, n104, n162, C25, C31, D50, D56, D90, D96
\@multiput I86, I95, I98
\@multispan H371, H375, H379
\@namedef f50, n384, r135, r136, r160, s7, s448, t372, A78, A114, A778, C28, D197, D207, D234, D260, D469, D478, E412, E413, H175, J12, J13, J18, J19, J23, J24, J25, P929, P1060, Q418, U378, U379
\@nameuse f51, n336, n369, n383, J23, O5, Q421, S607, S665
\@nbitem F168, F221
\@ne *155*, b16
\@needsf@rmat P569, P572, P577
\@needsformat P557, P567, P571
\@negargfalse I186
\@negargtrue I185
\@newcommand f79, f80
\@newctr p13, p15, J8
\@newenv f150, f151, f160
\@newenva f148, f149
\@newenvb f150, f151
\@newl@bel C22, D23, D78, N10
\@newline l94, l96
\@newlistfalse F29, F33, F108, F182, S600, S658
\@newlisttrue F29, F33, F87
\@next b275, L60, L129, L319, L320, S9, S213, S311, S877, S897, S1812
\@nextchar H269, H270, H330, H331, H332
\@nil a161, a162, c13, c19, c83, c84, f53, f54, f58, f63, f135, f439, f440, f585, f671, f672, i13, i19, i27, m14, n248, n249, n250, o89, o110, o979, o983, o1037, o1049, o1051, r329, r340, r456, r471, r575, r578, r579, r587, s393, s395, s528, s530, t304, t305, t307, t320, t326, t330, t331, t367, t388, t393, t473, t487, u26, u44, u53, u57, v40, v388, v396, v429, v1045, v1047, z58, z62, H367, H368, P79, P80, P86, P92, P97, P113, P171, P172, P183, P184, P195, P196, P198, P351, P479, P483, P489, P493, P637, P646, P660, P661, P1335, P1340, P1343, P1345, P1346, P1361, P1381, P1398, P1452, P1474, P1498, Q164, Q171, Q305, Q347, Q349
\@nmbrlistfalse F33, F46, F91
\@nmbrlisttrue F225
\@nnil f417, f423, i13, i20, i21, i22, i28, n249, n250, r214, r218, r219, r220, r235, t133, t135, t299, t301, t313, t315, t320, t334, t336, t343, t354, t355, t357, t388, t393, P609, P610, P617, I13
\@no@font@optfalse u17, u129
\@no@lnbk l9, l10, l40
\@no@pgbk l7, l8, l32
\@nobreakfalse l120, l122, F193, K94, K129, K157, L182, S165, S192, S1150, S1316
\@nobreaktrue l121, K126, L181
\@nocnterr j191
\@nocounterr j191, p4, p8, p16, p62, p64, J21
\@nodocument j196, n70, n138, n193, D155, L39, L108, S156, S183, S212
\@noitemargfalse F32, F200
\@noitemargtrue F32, F143
\@noitemerr j228, l248, l266, l303, l326, F69, F81, F107
\@noligs D418, D439, D547, D557, D568
\@nolnerr . . j189, l42, l113, D310, D317
\@nomath r1, r308, w401, w436, w442, w463, w465, w487
\@noparitemfalse F30, F145
\@noparitemtrue F30, F66
\@noparlistfalse F31, F70
\@noparlisttrue F31, F67
\@normalcr l52, l92, G303
\@normalsize P4, P5
\@noskipsec *497*

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@noskipsecfalse n56, n125, n180, K98, S158, S185
\@noskipsectrue K38, K95
\@notdefinable f136, f137, f141, f466, j183
\@notprerr j231, n68, n136, n191
\@nthm J3, J4
\@nxttabmar H11, H21, H23, H25, H75, H111, H112, H118, H119
\@obsoletefile n599
\@oddfoot O11, O14, O15, S124, S610, S669
\@oddhead . O11, O14, S123, S610, S669
\@onefilewithoptions *685*, P601, P605, P611, P629, P633, P639, P655, P656, P723, P783, P1311
\@onelevel@sanitize . f697, L42, L111
\@onfilewithoptions *659*
\@onlypreamble f66, f188, f190, f199, f207, g948, n198, n207, n244, n601, n627, o23, o24, o61, o62, o66, o125, o145, o189, o190, o204, r17, r115, r117, r123, r139, r167, r182, r203, r208, r250, r483, t373, u28, u36, u42, u79, u83, u88, u93, u98, u108, u126, u127, u128, u134, u138, u142, v17, v19, v44, v46, v107, v116, v136, v275, v276, v279, v311, v349, v351, v353, v366, v381, v428, v430, v472, v511, v527, v604, v698, v707, v763, v766, v769, v789, v802, v856, v891, v902, v916, v970, v990, v994, v1058, z140, z141, A79, A80, A81, A779, C30, M12, M29, N40, P10, P12, P32, P33, P74, P94, P102, P104, P127, P130, P131, P138, P141, P142, P150, P151, P152, P165, P192, P204, P205, P208, P247, P261, P324, P330, P373, P376, P377, P388, P389, P390, P397, P403, P416, P429, P447, P467, P472, P496, P501, P505, P508, P516, P521, P524, P534, P553, P566, P571, P577, P586, P591, P651, P723, P841, P850, P858, P859, P877, P884, P893, P900, P901, P909, P914, P919, P1291, P1292, P1293, P1294, P1296
\@opargbegintheorem J32, J35
\@opcol S262, S270, S394, S413, S442, S460, S465
\@options P466
\@othm J3, J20
\@outerparskip F8, F88, F117, F152, F222
\@outputbox S120, S483, S485, S505, S508, S509, S529, S531, S532, S537, S540, S545, S547, S554, S560, S562, S636, S695, S723, S729, S739, S740, S763, S770, S856, S859, S862, S868, S869, S2203, S2207, S2208, S2222, S2228, S2248, S2254, S2263
\@outputdblcol S468, S2198, S2200, S2244, S2245
\@outputpage S403, S452, S470, S590, S2232, S2237, S2270, S2278
\@oval I453, I471
\@ovbtrue I476, I504, I534
\@ovdx I431, I487, I489, I495, I497, I516, I518, I526, I528, I543, I551, I553, I589, I592, I599, I601, I693, I694, I695, I696, I712, I713, I714, I717, I733, I753, I754, I755, I756, I772, I773, I775, I789
\@ovdy I431, I488, I490, I496, I497, I517, I519, I527, I528, I544, I552, I553, I564, I569, I577, I581, I700, I701, I702, I703, I718, I719, I720, I723, I737, I760, I761, I762, I763, I776, I777, I779, I793
\@ovhlinefalse I477, I505
\@ovhlinetrue I456, I460, I465, I483, I489, I511, I518
\@ovhorz I494, I495, I525, I526, I550, I551, I584
\@ovltrue I476, I504, I534
\@ovri G33, I431, I486, I515, I542, I564, I577, I593, I602
\@ovro I431, I486, I495, I496, I515, I526, I527, I542, I551, I552, I563, I569, I576, I581, I588, I598, I614, I621, I630, I639
\@ovrtrue I476, I504, I534
\@ovttrue I476, I504, I534
\@ovvert I492, I493, I521, I523, I546, I548, I557
\@ovvlinefalse I477, I505
\@ovvlinetrue I459, I482, I490, I510, I519
\@ovxx I431, I480, I482, I483, I487, I493, I494, I508, I510, I511, I516, I523, I525, I537, I539, I543, I548, I550, I588, I598, I690, I691, I692, I696, I705, I706, I715, I716, I717, I732, I750, I751, I752, I756, I765, I766, I774, I775, I788
\@ovyy I431, I481, I482, I483, I488, I495, I509, I510, I511,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

I517, I526, I538, I539, I544, I551, I561, I574, I697, I698, I699, I703, I705, I721, I722, I723, I736, I757, I758, I759, I763, I765, I778, I779, I792
\@p@pfilename P79, P86, P92, P97, P102
\@p@pfilepath P80, P111, P120
\@p@pfilepath@aux . . P112, P113, P121
\@pagedp S117, S308, S313, S1091, S1244, S1841, S1851
\@pageht S116, S309, S313, S315, S316, S317, S321, S1090, S1243, S1824, S1831
\@par k3, k5
\@parboxrestore G247, G303, G328, G355, L19, L100, L169, L338, L442, S219, S601, S659
\@parboxto G242
\@parmoderr j221, L58, L127, L316
\@parse@version c83, c84, P177, P183, P184, P195, P1346, P1361, P1398, P1474, P1498
\@parse@version@ P171, P172, P177, P189
\@parse@version@dash . . . P196, P198
\@partaux n5, n224, n268, n286, n309, n311, n312, n332, n356, n358, n359, n365, n377, n379, n382
\@partlist n233, n237, n238, n240, n264, n283, n305, n352
\@partswfalse n8
\@partswtrue n232, n262, n282
\@pass@ptions P346, P347, P349, P364, P365, P366, P373, P374, P375, P734, P809
\@pboxswfalse G245, G321
\@pboxswtrue G255
\@penup E186, E187
\@percentchar a106, P1006, P1008, P1010, P1012, P1137, P1139, P1141, P1143, P1226, P1228, P1230, P1232, P1280
\@picbox I6, I31, I43, I51, I52
\@picht I6, I29, I42, I51
\@picture I23, I24
\@picture@warn . I223, I441, I445, I449
\@pkgextension *657*, P30, P128, P139, P202, P374, P507, P510, P533, P542, P612, P640, P742, P769, P895, Q418, Q424
\@plus f26, l456, K16, K191, K214, O54, S2306, S2307, S2308, S2311, S2312, S2316, S2317, S2318, S2322, S2323, S2324
\@pnumwidth K203, K226
\@popfilename *639*, *639*, *656*, P34, P720, P831
\@pr@videpackage P260, P274, P277, P298, P300, P311, P313, P324
\@preamble H190, H192, H200, H237, H256, H258, H259, H263, H278, H296, H297, H334
\@preamblecmds f66, n69, n137, n192, P1512, P1513
\@preamerr . . . j210, H199, H274, H355
\@process@pti@ns P415, P428, P433, P434, P447, P451, P453
\@process@ptions . . . P402, P404, P416
\@protected@testopt f89, f101, f580, f592
\@providesfile . a98, a99, P334, U539
\@ptionlist P125, P207, P401, P672, P678, P799, P805, P896
\@pushfilename *639*, *639*, P34, P662, P786
\@put . I452, I497, I528, I553, I621, I639
\@qend f136, f671, j187
\@qrelax f137, f671
\@rc@ifdefinable . f130, f132, f238, o14
\@reargdef f122
\@refundefined n57, n126, n181, C3, D48, D88
\@reinserts S327, S330, S516
\@remove@tlig o979, o987
\@remove@tlig@ o979, o980
\@remove@tlig@@ o980, o983
\@remove@tlig@@@ o1013
\@remove@tlig@@@@ o993
\@removeelement i32, P469
\@removefromreset p46, p69, p72, p91, p92
\@reqcolroom S1090, S1091, S1094, S1096, S1097, S1102, S1106, S1108, S1136, S1137, S1243, S1244, S1248, S1251, S1252, S1257, S1264, S1266, S1298, S1299, S1410, S1412, S1414, S1417, S1419, S1493, S1496, S1499, S1504, S1506, S1986, S2103, S2108, S2111
\@reserveda *304*
\@reset@ptions P684, P721, P732, P790, P832, P842
\@resetactivechars . S575, S598, S656
\@resethfps S1205, S1374, S2053
\@restorepar k5, l341, l357, l375, l391, F127, F135
\@reversemarginfalse L364, S101
\@reversemargintrue L363
\@rightmark O16, O51

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@rightskip D349, D368, D386, F75, G276, G297
\@rjfieldfalse H34, H77
\@rjfieldtrue H125
\@rmfamilyhook w261, w304, w340
\@roman p103, p109
\@rsbox G395, G402, G406
\@rtab H71, H86
\@rule G367, G372, G376
\@sanitize . . f695, M7, M18, M24, M35
\@savebox G103, G110, G116
\@savemarbox . . L326, L327, L330, L333
\@savepicbox G103, G110, G120
\@savsf l123, l129, l138, l157, l171, l183, l197, l211
\@savsk l123, l128, l139, l158, l172, l184, l198, l212
\@scolelt S788, S853
\@sdblcolelt S805, S825, S854
\@seccntformat K60, K111
\@secondoftwo a88, e17, e21, f212, f445, f539, f589, f640, f647, f667, f714, n425, n463, o131, p156, p161, w362, A92, A747, A798, A814, C21, O17, P134, P173, P185, P222, P240, P1503
\@secpenalty l17, K36, K50
\@sect K54, K55
\@seqncr E411
\@set@curr@file@aux Q299
\@setckpt n377, n384, D22, D77
\@setfloattypecounts S1074, S1224, S1395, S1476, S1571, S1692, S2000
\@setfontsize w487
\@setfps L34
\@setfpsbit L73, L75, L77, L85, L143, L146, L149, S2044
\@setmarks S2214, S2216, S2231
\@setminipage G333, L21, L177, L185, L352
\@setnobreak L179, L351
\@setpar k3, F78
\@setref C10
\@setsize w487
\@settab H71, H93
\@settodim q17
\@settopoint q22
\@setupverbvisiblespace D459, D470, D482, D499, D514
\@sffamilyhook w261, w309, w341
\@sharp H196, H235, H265, H280, H281, H301, H303, H305, H333
\@shipoutsetup S590
\@shortstack I132, I133
\@show@... . . . . *90*
\@show@DeclareRobustCommand f507, f521, f566, f617, f620
\@show@newcommand *91*, f508, f521, f574, f600, f617, f623
\@show@newcommand@aux . *92*, f600, f627
\@showcommandlisthook *89*, *90*, f504, f506, f514
\@sline I167, I179, I184, I269
\@slowromancap p110, p111
\@spaces j169
\@specialoutput S256
\@specialpagefalse . . S97, S607, S665
\@specialpagetrue O9
\@specialstyle O9, S607, S665
\@sptoken f680, f690
\@sqrt E342
\@ssect K53, K112
\@stackcr I139, I142
\@star@or@long f72, f77, f124, f146, f152, f178, f187, f221
\@startcolumn S263, S270, S775
\@startdblcolumn S775, S2236, S2239, S2276, S2282
\@startfield H28, H46, H92, H104, H125, H133
\@startline H20, H57, H62, H68, H83, H154
\@startpbox H193, H236, H266, H332, H384, H386
\@startsection K39
\@starttoc K149
\@stopfield H32, H48, H86, H93, H125, H127, H140, H142, H154
\@stopline H30, H56, H85
\@stpelt p20, p23
\@string@makeletter *96*, f701
\@strip@args o76
\@strip@tex@ext *201*, n228, n238, n240, n245, n275
\@strip@tex@ext@aux n245, n276
\@svector I244, I259, I269
\@sverb . *478*, D486, D549, D558, D561
\@svsec K57, K60, K66, K78
\@svsechd K76, K101, K121
\@swaptwoargs n480, n482, P694, Q157, Q180
\@sxverbatim D394, D471, D478
\@tabacckludge o223, o225, o451, o452, A197, A204, A206
\@tabacol H178, H258
\@tabarray H170, H180, H181
\@tabclassiv H180, H330
\@tabclassz H179, H282

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@tabcr H56, H73
\@tabfbox H16, H80, H82
\@tablab H72, H126
\@tabminus H72, H117
\@tabplus H72, H110
\@tabpush H11, H77, H85, H140, H143, H145
\@tabrj H72, H124
\@tabular H174, H177, H178
\@tabularcr H180, H208
\@tempa C47, C48, C59, C60
\@tempboxa h13, o69, q17, q18, F205, F211, F212, F214, G29, G30, G31, G32, G37, G38, G39, G40, G156, G186, G193, G203, G324, G347, G412, G413, G414, G421, G422, G423, G424, K138, K139, L322, L356, S305, S377, S382, S383, S424, S429, S430, S566, S626, S633, S634, S685, S692, S693, S721, S725, S737, S743, S750, S751, S752, S753, S757, S765, I304, I305, I447, I448, I486, I491, I496, I497, I515, I520, I527, I528, I542, I545, I552, I553, I614, I615, I620, I621, I630, I631, I638, I639, I724, I742, I780, I798
\@tempcnta h7, v771, v772, v773, v774, v778, H242, H243, H244, H245, L62, L68, L70, L79, L80, L90, L91, L131, L137, L139, L152, L153, L159, L160, P952, P954, P955, P956, P1083, P1085, P1086, P1087, P1197, P1199, P1200, P1201, S16, S18, S20, S934, S935, S936, S937, S957, S958, S959, S960, S982, S985, S1018, S1021, S1132, S1294, S1574, S1577, S1695, S1698, S1813, S1815, S1818, S1820, S1822, S1844, S2034, S2035, S2039, S2045, S2049, U160, U165, U166, U167, U262, U264, U265, U266, U273, U275, U276, U277, U283, U285, U286, U287, U294, U296, U297, U298, U324, U326, U327, U328, U350, U352, U353, U354, U360, U362, U363, U364, U393, U398, U399, U400, I187, I188, I214, I215, I216, I229, I230, I231, I232, I239, I240, I254, I255, I270, I271, I276, I278, I279, I280, I281, I282, I285, I287, I288, I289, I290, I291, I292, I293, I294, I295, I296, I335, I336, I337, I338, I339, I360, I361, I362, I363, I364, I365, I392, I393, I394, I395, I396, I414, I416, I418, I419, I421, I423, I438, I439, I440, I442, I444, I446, I448, I562, I567, I575, I579, I616, I617, I618, I619, I632, I633, I635, I636, I658, I659, I660, I661, I662, I663, I711, I731, I771, I787
\@tempcntb . . . h7, v772, v776, v778, L88, L89, L90, L157, L158, L159, S17, S20, S21, S2045, S2046, S2047, U161, U165, U394, U398, I279, I280, I281, I283, I284, I285, I562, I563, I567, I568, I575, I576, I579, I580
\@tempdima h10, r219, r224, E173, E176, E182, G43, G44, G185, G186, G191, G192, G193, G195, G246, G247, G322, G326, G379, G382, G383, G410, G412, G418, G421, H35, H36, H37, H88, H89, H90, H91, H218, H219, K196, K197, K219, K220, K233, L196, L198, L218, L220, L258, L259, L260, S229, S230, S231, S487, S489, S535, S537, S538, S543, S548, S552, S557, S561, S917, S920, S940, S950, S962, S972, S1637, S1638, S1641, S1642, S1762, S1763, S1767, S1768, S1823, S1824, S1825, S1826, S1829, S1832, S1835, S1837, S2152, S2153, S2155, S2156, I210, I211, I213, I214, I215, I216, I217, I218, I437, I438, I439, I448, I487, I488, I492, I493, I516, I517, I521, I523, I543, I544, I546, I548, I617, I619, I634, I636, I637, I657, I658, I659
\@tempdimb h10, r220, r225, r595, r599, t133, t134, t391, t414, t415, t424, t425, t429, t447, t450, t453, t455, G249, G250, G380, G383, G411, G413, G419, G422, S940, S941, S942, S943, S950, S962, S963, S964, S965, S972, I211, I212, I357, I359, I362, I365, I482, I484, I485, I510, I513, I514, I539, I540, I541, I612, I613, I622, I628, I629, I640, I648, I649, I654
\@tempdimc h10, t408, t409, t411, t412, t414, t415, G53, G54, G64, G65, G381, G382, G383, I30, I31, I32, I33, I34, I35, I60, I61, I63, I64, I332, I333, I334
\@tempskipa h14, l44, l47, l48, l285, l292, l294, l297, t135, t136, F116, F117, F118, F150, F152,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

F153, F154, F222, F223, F224, K42, K44, K45, K50, K62, K63, K88, K89, K91, K103, K104, K113, K114, S1872, S1873, S1875, S1883
\@tempskipb *185*, h14, l220, l222, l224, l227, l229, l243, l261, l283, l285, l286, l290, l292, l294, l295, l318, l321
\@tempswafalse a78, b262, n303, n350, o1497, r94, r644, v313, v368, v432, v513, v1033, v1039, D24, D79, D409, D430, N13, P929, P945, P1060, P1076, Q268, S988, S1024, S1580, S1701
\@tempswatrue a79, b268, n301, n306, n348, n353, o1500, o1501, r97, r645, r646, r649, r652, v316, v371, v435, v516, v996, D110, D414, D435, N13, P926, P1057, Q268, S1582, S1605, S1703, S1728, S2113, S2130
\@temptokena *465*, h16, D122, D126, D135, D148, D149, O26, O27, O34, O35, O48, O49
\@testdef D23, D78, D108
\@testfalse S12, S14, S15
\@testfp S882, S902, S938, S961, S2037, S2166, S2183
\@testopt f33, f79, f99, f103, f148, l7, l8, l9, l10, E382
\@testpach H270, H348
\@testpatch H348
\@testtrue S13, S21, S356, S885, S904, S944, S966, S2041
\@testwrongwidth S345, S883, S939, S1112, S1426, S1631
\@text@composite . . . o76, o1044, o1049
\@text@composite@x o76
\@textbottom O54, O56, S511, S549, S563, S572
\@textfloatsheight . . S476, S1087, S1089, S1139, S1140, S1145, S1240, S1242, S1302, S1304, S1310, S1986
\@textmin . L285, L286, L299, L300, S112, S1089, S1093, S1096, S1097, S1242, S1247, S1251, S1252, S1414, S1499, S1598, S1600, S1616, S1720, S1722, S1740, S2094, S2096, S2098
\@textsubscript L400, L408
\@textsuperscript . . L377, L379, L380
\@texttop . O54, O56, S507, S530, S572
\@tf@r i25, i26
\@tfor i25, n426, n464, n608, z88, G57, G76, H268, L63, L132, I478, I506, I535
\@tforloop i27, i28, i30
\@thanks K11, K34
\@thefnmark G357, L376, L377, L429, L434, L444, L453, L458, L469, L474
\@thefoot S124, S610, S613, S640, S669, S672, S699
\@thehead S123, S610, S612, S630, S669, S671, S689
\@themargin S74, S611, S613, S625, S670, S672, S684
\@themark O25, O26, O33, O34, O49, O52
\@thirdofthree f216, o197
\@thm J12, J18, J24, J26
\@thmcounter J11, J17, J33
\@thmcountersep J10, J33
\@title K7, K31
\@tocrmarg K192, K215
\@toodeep j203, F36, F232, F243
\@toplist S64, S384, S385, S431, S432, S716, S722, S732, S733, S1025, S1037, S1921, S1948
\@topnewpage S199
\@topnum L271, S105, S1022, S1023, S1037, S1041, S1458, S1463, S1551, S1558, S1912, S1939, S1980
\@toproom L273, S106, S1025, S1037, S1913, S1940
\@topsep F1, F71, F73, F171
\@topsepadd . . F1, F59, F61, F71, F124
\@totalleftmargin *497*, D405, D427, F9, F53, F54, G275, G296, H35, H76, H81
\@trivlist F48, F57, F92
\@tryfcolumn S778, S798, S816, S832, S2167, S2184
\@trylist . S841, S844, S877, S897, S919
\@ttfamilyhook w261, w314, w342
\@twoclasseserror P498, P915
\@twocolumnfalse S99, S147
\@twocolumntrue S206
\@twoloadclasserror P719, P830, P910
\@twosidefalse S100
\@typein f32, f33, f40, f48
\@typeset@protect *81*, f102, f243, f250, f252, o26, o32, o210, o218, w488, D182, D227, D245, U256
\@uclclist . o1466, o1467, o1520, U463
\@undefind I57, I71, I82, I91, I162, I174, I237, I252, I313, I371
\@undefined a68, a69, a108, a109, a110, a131, a139, a147, a154, a205, a209, a235, a242, a326, a327, b65, b81, b105, b106, b121, b122, b127, b136, b149, b184, b189, b222, b223, b246, b256,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

b291, b459, b502, b548, b549, c49, c57, d2, d15, d16, d17, d29, d30, d74, d84, d173, d181, d189, d197, d229, d230, d231, d232, d233, d234, d235, d236, d237, d238, d239, d240, d241, d242, d243, d244, d245, d246, d247, d253, e143, e144, e145, e146, e147, e148, f34, f223, f307, f308, f339, f367, f371, f372, f389, f403, f450, f451, f452, f453, f454, f455, f456, f490, f491, f492, f493, f494, f495, f513, f514, f618, f619, f620, f621, f622, f623, f625, f626, f627, f642, f649, f723, f724, f725, f734, j28, l89, l470, l476, n63, n64, n131, n132, n186, n187, n275, n276, n413, n441, n442, n443, n445, n450, n491, o195, o197, o333, o335, o337, o339, o341, o343, o345, o347, o349, o351, o370, o372, o374, o455, o694, o697, p170, r405, r445, r507, r540, r604, r611, s452, s457, s462, s583, s584, s586, s587, s588, s589, s590, s591, s592, s593, s594, s595, s596, s597, s598, s599, s600, s601, s602, s603, s604, t506, t507, u4, u5, u6, u7, u8, u9, u10, u11, u12, u13, u14, u15, u16, u17, u18, u19, u20, v629, v736, w35, w55, w166, w167, w168, w169, w170, w171, w172, w173, w174, w175, w176, w178, w179, w180, w322, w340, w341, w342, w384, w385, w386, w387, w388, w428, w429, w430, w431, w432, w433, w444, w529, x15, x50, x65, z37, z38, z39, z122, A52, A179, A294, A536, A537, A567, A568, A569, A571, A572, A573, A574, A575, A576, A579, A580, A581, A582, A585, A586, A587, A588, A590, A591, A592, A594, A595, A600, A601, A602, A603, A604, A605, A606, A607, A608, A609, A610, A611, A612, A613, A614, A615, A617, A618, A620, A621, A622, A623, A624, A625, A626, A627, A628, A629, A631, A632, A633, A634, A635, A636, A637, A638, A639, A641, A642, A643, A644, A645, A646, A647, A648, A649, A650, A651, A652, A653, A654, A655, A656, A657, A658, A659, A660, A661, A662, A663, A664, A665, A666, A667, A672, A673, A675, A676, A677, A678, A679, A680, A681, A682, A684, A685, A687, A688, A690, A691, A692, A693, A695, A696, A697, A699, A701, A702, A703, A704, A705, A706, A707, A709, A710, A711, A712, A713, A714, A715, A716, A717, C53, C69, C70, D102, D103, D104, D142, D143, D144, D145, D298, D299, D300, D301, D453, D480, D481, D482, D483, D516, E223, E233, E234, E270, E273, E316, E329, G21, G111, G175, G231, G373, G403, K230, L5, L405, L424, N33, P4, P24, P119, P120, P121, P189, P320, P724, P759, P761, P767, P820, P835, P836, P851, P965, P1044, P1047, P1096, P1175, P1178, P1188, P1189, P1190, P1191, P1192, P1193, P1194, P1265, P1268, P1284, P1304, P1311, Q18, Q19, Q20, Q21, Q144, Q182, Q183, Q190, Q191, Q251, Q252, Q296, Q424, Q430, Q431, Q432, R352, R353, R354, R355, R357, R358, R359, R366, R367, R369, R371, R372, R373, R375, R377, R378, R379, R380, R384, R407, S36, S368, S369, U10, U18, U25, U40, U59, U68, U75, U93, U94, U203, U216, U228, U249, U250, U251, U280, U282, U321, U322, U341, U342, U343, U344, U345, U346, U347, U348, U349, U366, U382, U383, U384, U432, U433, U493, U528, U529, U530, U531, U532, I18, I467, I468

`\@unexpandable@protect` f220, f255, f261, f266, n212, H264

`\@unknownoptionerror` P846, P885, P898

`\@unprocessedoptions` *657*, *658*, P463, P541, P724, P760, P761, P817, P821, P900

`\@unused` f10, f17, j15, j32, j59, n3, P1289

`\@unusedoptionlist` . n20, n22, n89, n91, n146, n148, P11, P393, P394, P470

`\@upline` I297, I298, I304

`\@upordown` . I195, I196, I204, I225, I273

`\@upvector` I268, I304

`\@use@ption` P411, P423, P437, P439, P456, P458, P468

`\@use@text@encoding` o150, A16, A1050

`\@vbsphack` l219

`\@verb` D549, D558, D561

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

\@verbatim . . . D399, D445, D469, D478
\@verbvisiblespacebox D464, D465, D468, D483
\@vereq x457, x458
\@viiipt r687
\@viipt r686
\@vipt r685
\@vline I166, I178, I297
\@vobeyspaces D392, D445, D471, D499, D514, D561
\@vpt r684
\@vspace l330
\@vspace@calcify l79, l101, l241, l337, l342, l352, l360, D328, E390, H62, H224, I148
\@vspacer l330
\@vtryfc S847, S855
\@vvector I243, I258, I268
\@warning j166
\@wckptelt n378, n381
\@whiledim i7, I123, I203
\@whilenoop i3
\@whilenum i3, H244, I104, I116, I336, I338, I361, I364, I393, I395, I416, I421, I731, I787
\@whilesw i10, S264, S394, S403, S441, S451, S2237, S2277
\@whileswnoop i10
\@wholewidth G141, G143, G144, G146, G148, G149, G150, G151, I2, I126, I129, I131, I299, I302, I350, I359, I406, I413, I565, I578, I590, I600, I679, I680, I728, I784
\@width b405, f26, l452, o289, o294, t146, x618, G146, G148, G199, G206, G383, G427, H188, H219, H347, H366, L371, S1851, S2226, S2260, I227, I299, I302, I325, I333, I350, I359, I384, I392, I406, I413, I565, I578, I728, I784
\@wrglossary M25, M30
\@wrindex M8, M13
\@writeckpt n330, n363, n375
\@writefile *464, 465, 465, 465, 597*, n34, n103, n161, D129, K183
\@writesetup S590
\@wrong@font@char o161, r508, r542, r555
\@wtryfc S857, S867
\@x@protect f105, f242
\@x@sf L463, L465
\@xDeclareMathDelimiter . . v801, v857
\@xaddvskip l219, l244, l262
\@xarg . I163, I166, I175, I178, I185, I189, I190, I226, I228, I238, I239, I243, I253, I254, I258, I266, I274, I664
\@xargarraycr H205, H214, H218
\@xargdef f80
\@xarraycr H202, H203
\@xbitor S15, S17
\@xcentercr D311, D318, D322
\@xdblarg f693
\@xdblfloat L264
\@xdim I84, I93, I105, I107, I117, I119, I668, I732, I733, I734, I735, I741, I788, I789, I790, I791, I797
\@xeqncr E374
\@xexnoop H238, H248
\@xexpast H239, H240
\@xfloat L28, L29, L34, L266
\@xfootnote L428, L431
\@xfootnotemark L451, L455
\@xfootnotenext L468, L471
\@xhline H360, H361
\@xifnch f681, f691
\@xiipt r691, x164, x166, x167
\@xipt r690, x163
\@xivpt r692, x165, x167
\@xmpar L324, L325
\@xnewline l60, l61, l72, l73, l93
\@xnext S10, S11
\@xnthm J5, J6
\@xobeysp *189*, l420, D393, D394, D461, D465
\@xprocess@ptions . . P402, P417, P429
\@xpt r689, x162, x165, x166
\@xsect K86, K87, K123
\@xtabcr H56, H57
\@xtabularcr H209, H210
\@xthm J28, J29
\@xtryfc S844, S872
\@xtypein f33, f35, f42
\@xverbatim D394, D445
\@xviipt r693, x166, x168
\@xxDeclareMathDelimiter . v786, v790
\@xxpt r694, x167, x168
\@xxvpt r695, x168
\@xxxii h2, o422, o424, L89, L158, S879, S880, S899, S900, S935, S936, S958, S959, S2003
\@xympar L328, L332, L354
\@yarg I163, I167, I175, I179, I185, I186, I195, I238, I244, I253, I259, I268, I270, I297, I664
\@yargarraycr H206, H216, H220
\@yargd@f f107
\@yargdef f84, f94, f107, f123

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\@ydim I85, I94, I105, I108, I117, I119, I668, I736, I737, I738, I739, I740, I792, I793, I794, I795, I796
\@yeqncr E374
\@ympar L324, L329
\@ynthm J5, J14
\@ythm J28, J29
\@ytryfc S890, S909, S913
\@yyarg I185, I186, I187, I190, I274, I664
\@ztryfc S918, S929
\@@end *118*
\accent@spacefactor . . . o70, o73, o74
\active@math@prime . E240, E241, S588
\add@accent o65, o67
\add@percent@to@temptokena *466*, D116, D132, D134, D143
\add@unicode@accent . . . o1023, o1027
\addto@hook . r152, r154, r683, v295, v391, v395, v412, v536, v542, v550, v566, v569, v572, v1005, v1012, v1015
\alloc@ b90, b91, b92, b93, b94, b95, b96, b97, b98, b99, b226, d22, d26, d38, r14
\alpha@elt v45, v299, v486, v588, v1004, v1005
\alpha@list v41, v43, v308, v474, v486, v531, v586, v587, v1000, v1006, v1007
\atveryend@DEPRECATED . . Q507, Q509
\best@size t392, t416, t422, t428
\bf@def@ult w190
\bfdef@ult *305*, *375*, *382*, w135, w194, w205, w206, w207, w274
\bfdefault@previous *375*, *375*, w201, w204, x106, x116
\bfseries@.. . . . . *370*
\bfseries@rm . . *370*, *373*, *381*, w63, w142, w145, w167, w205, w210, w282
\bfseries@rm@kernel *373*, w66, w142, w170
\bfseries@sf *368*, *373*, w63, w146, w149, w168, w206, w211, w283
\bfseries@sf@kernel w67, w146, w171
\bfseries@tt *373*, w63, w150, w153, w169, w207, w212, w284
\bfseries@tt@kernel w68, w150, w172
\bm@b G37
\bm@c G37
\bm@l G37
\bm@r G37
\bm@s G37
\bm@t G37
\bx@A S30, S57
\bx@AA S40
\bx@B S30, S57
\bx@BB S40
\bx@C S30, S57
\bx@CC S40
\bx@D S30, S57
\bx@DD S40
\bx@E S30, S57
\bx@EE S40
\bx@F S31, S58
\bx@FF S41
\bx@G S31, S58
\bx@GG S41
\bx@H S31, S58
\bx@HH S41
\bx@I S31, S58
\bx@II S41
\bx@J S31, S58
\bx@JJ S41
\bx@K S32, S59
\bx@KK S42
\bx@L S32, S59
\bx@LL S42
\bx@M S32, S59
\bx@MM S42
\bx@N S32, S59
\bx@NN S42
\bx@O S33, S60
\bx@OO S43
\bx@P S33, S60
\bx@PP S43
\bx@Q S33, S60
\bx@QQ S43
\bx@R S33, S60
\bx@RR S43
\bx@S S38
\bx@SS S44
\bx@T S38
\bx@TT S44
\bx@U S38
\bx@UU S44
\bx@V S38
\bx@VV S44
\bx@W S39
\bx@WW S45
\bx@X S39
\bx@XX S45
\bx@Y S39
\bx@YY S45
\bx@Z S39
\bx@ZZ S25, S45, S55
\c@bottomnumber . . L269, L274, S2292
\c@dbltopnumber L268, L283, L297, S2299
\c@enumi F227
\c@enumii F227

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\c@enumiv F227
\c@equation E336, E369, E511
\c@errorcontextlines j163
\c@footnote K12, L373, L457
\c@mpfootnote G329, L375
\c@ncel x461, x462
\c@page B3, B6, B7, S138, S1818
\c@secnumdepth . K56, K71, K81, K140
\c@tocdepth K140, K190, K213
\c@topnumber L267, L271, S2288
\c@totalnumber . . . L270, L276, S2295
\c@totalpages R272, R354
\calculate@math@sizes . . . r591, t173
\catcodetable@atletter d93, d238
\catcodetable@initex d93, d235
\catcodetable@latex d93, d237
\catcodetable@string d93, d236
\cdp@elt r96, r116, r127, r128, r149, r152, r154, v233, v315, v370, v434, v515, U370, U371
\cdp@list r98, r114, r128, r156, r157, v251, v317, v372, v436, v517, U371
\cf@encoding o34, o41, o44, o51, o154, r256, r266, r276, r297
\ch@ck b206, b207, b208, b209, b236, b248, b249, b250, b251, b279, b281, b293, b294, b295, b296, b302, P970, P991, P1101, P1122, P1214
\char@if@alph f701
\chardef@text@cmd o3
\check@command f187, f189
\check@icl z9, z44, z49, z55, z63, z70, z72
\check@icr z9, z44, z50, z56, z64, z73, z78
\check@mathfonts . . m5, o302, o328, o360, o1220, r319, r321, t204, A584
\check@nocorr@ z46
\check@range t333, t334
\check@single t332, t354
\cl@@ckpt n378, p35
\cl@page B4
\col@number S95, S148, S208, S220
\color@begingroup r614, r674, E104, E134, G29, G89, G157, G325, G358, H47, H51, L446, S491
\color@endbox G89, L253, L344, S224, S631, S641, S690, S700
\color@endgroup r619, r680, E104, E134, G29, G89, G115, G136, G159, G345, G361, H49, L449, S495
\color@hbox G89, S628, S638, S687, S697
\color@setgroup . . . G89, G115, G134
\color@vbox G89, L96, L165, L335, L357, S215
\conditionally@traceoff . . . *170*, j239
\conditionally@traceon j239
\contionally@traceon *170*
\copy@kernel@robust@command f541, f625
\count@ a66, a179, a180, a181, a186, b41, b191, b192, b197, b199, b205, b206, b207, b208, b209, b210, b405, b406, c14, c15, c16, c17, c18, c20, f169, f173, f275, f300, r648, r654, r656, t22, t256, t258, t280, t281, v292, v294, v298, v617, v618, v619, v665, v666, v667, v726, v727, v728, v774, v775, v776, v814, v815, v816, v822, v823, v824, v868, v869, v870, v876, v877, v878, v937, v938, v939, v945, v946, v947, z115, z118, P1015, P1017, P1018, P1019, P1146, P1148, P1149, P1150, P1235, P1237, P1238, P1239, U172, U173, U180, U182, U437, U438, U445, U447, I730, I731, I732, I735, I736, I739, I743, I786, I787, I788, I791, I792, I795, I799
\counterwithin@s p77, p78, p98
\counterwithin@x p77, p80, p99
\counterwithout@s p67, p68, p95
\counterwithout@x p67, p70, p96
\curr@fontshape o180, r88, r334, r342, r346, r348, r490, r496, r499, r508, r515, r517, r525, r531, r534, r542, r549, r551, s417, t92, t100, t121, t431, t451, t483, t499, t514, v255, v260, w406, w415
\curr@math@size r323, t210, t216, t221, t238
\declare@commandcopy *88*, f464, f468, f473, f476, f493
\declare@commandcopy@let *88*, f481, f485, f495, f563, f564
\declare@file@substitution *682*, Q238, Q494, R437, R468
\declare@robustcommand f221
\DeclareEncodingSubset@aux A62, A64, A80
\DeclareFontEncoding@ r122, r124, r139, U281, U301, U367
\DeclareFontEncoding@saved U281, U301, U383
\DeclareFontShape@ r21, r22
\DeclareSymbolFontAlphabet@ v992, v995
\default@ds P387, P398, P438, P457, P844, P846
\default@family r129, r161, r458, r472, r475, r500, r535, U372

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\default@M r136, r176, r179, r183, U379
\default@mextra u10, u89
\default@series r129, r162, r459, r473, r476, r497, r532, U372
\default@shape r130, r163, r460, r474, r477, r495, r530, U373
\default@T r170, r173, r183, r272
\define@mathalphabet u18, u131
\define@mathgroup u19, u135
\define@newfont r326, r335
\development@branch@name c11, c36, c49, c50, c51, c57, c58, c59
\dimen@ b41, b402, b403, b439, b440, b442, b444, j28, j29, l349, l354, l383, l388, o426, o427, o429, o430, o790, o791, r214, r216, r222, r235, r238, r242, r594, r595, r596, r600, t405, t406, t407, t408, t412, A123, A125, A840, A842, E72, E73, E186, E187, E188, E189, G420, G423, H176, H177, S508, S510, S531, S533
\dimen@i b41
\dimen@ii b41, r218, r223
\disable@package@load *683*, *696*, Q413, R396
\displ@y E191, E195, E196
\do@add@percent@to@temptokena D120, D126, D144
\do@emfont@update . . w407, w411, w431
\do@noligs D563, D568
\do@subst@correction . r84, t436, t491
\document@default@language n53, n54, n122, n123, S597, U219
\document@select@group . . . v137, v268
\dont@add@percent@to@temptokena D119, D121, D145
\dorestore@version v114, v119
\ds@ P400, P848
\dt@pfalse E192
\dt@ptrue E191
\e@alloc b51, b52, b53, b55, b56, b63, b64, b66, b68, b79, b82, b84, b138, b230, d15, d49, d79, d89, d178, d186, d194, d202, U12, U33
\e@alloc@attribute@count d66, d74, d75, d76, d80, d229
\e@alloc@bytecode@count d70, d189, d190, d191, d195, d245
\e@alloc@ccodetable@count d67, d84, d85, d86, d90, d233
\e@alloc@chardef . b60, b102, b210, b211, d48, d178, d186, d194, d202, U12
\e@alloc@intercharclass@top . . . U21
\e@alloc@luachunk@count d71, d197, d198, d199, d203, d247
\e@alloc@luafunction@count . d68, d173, d174, d175, d179, d239, d241
\e@alloc@top b55, b63, b102, b188, b259, d47, d80, d179, d187, d195, d203, U12
\e@alloc@whatsit@count d69, d181, d182, d183, d187, d243
\e@ch@ck b142, b152, d51, d55
\e@insert@top . b257, b259, b276, b291
\e@mathgroup@top b79, b124, v56, v145, v176
\em@currfont *384*, w406, w417
\em@force *384*, *384*, w415, w420, w423, w433
\emfontdeclare@clist . *383*, w396, w398, w402, w407, w412, w418, w429
\empty@sfcnt t444, t445, t446, t460, t465, t517
\enc@update r257, r259, r275, r278, t129
\end@dblfloat L205
\end@float *614*, L189, L227, L243, L359
\err@rel@i u12, u99, u132, u136
\error@fontshape r453, r468, r493, r528, t107, t481, v254
\et@xmaxfam d22, d26, d30, d38
\et@xmaxregs d29, d31, d32, d33, d34, d35, d36, d37
\every@math@size r78, t189, t201
\execute@size@function t316, t344, t358, t375
\expand@font@defaults *119*, w83, w190, w200, w220, w270, w281, w292, w322, w351, A26, A104
expand@font@defaults w252
\external@font t84, t87, t98, t102, t104, t345, t359, t421, t455, t523, t525, t527
\extra@def u9, u84
\extract@alph@from@version r568, r574, v151, v182, v214
\extract@default@composite o1036, o1043
\extract@default@composite@a o1045, o1049
\extract@default@composite@b o1047, o1051
\extract@font r349, t81
\extract@fontinfo t312, t319
\extract@rangefontinfo t329, t336, t355, t388
\extract@sizefn t304, t326

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\f@baselineskip o1186, r288, r295, r479, t119, t136, t140, t155, t169, t180, t194
\f@depth L291, S345
\f@encoding d252, d267, d275, o178, r251, r270, r273, r274, r276, r297, r329, r334, r353, r355, r357, r362, r364, r395, r457, r489, r524, s402, s406, s536, s540, t91, t261, t471, v239
\f@family *277*, *375*, *381*, r279, r283, r291, r330, r334, r353, r355, r357, r362, r364, r396, r475, r500, r535, s402, s406, s536, s540, t91, v239, v270, w86, w113, w114, w210, w211, w212, w229, w230, w231, w282, w283, w284, w293, w294, w295, w354, w367, w509, A23, A27, A29, A31, A84, A87, A101, A105, A107, A109, A114, A121, A806, A809, A823, A832, A838, A1053
\f@linespread r291, t118, t137, t138, t141, t149, t152, t163, t166
\f@series . . *294*, *294*, *303*, *303*, *304*, *305*, *308*, *371*, *372*, *373*, *377*, *380*, *381*, *382*, m14, r279, r284, r292, r331, r334, r476, r497, r532, s386, s391, s425, s426, s536, s540, s581, v271, w79, w95, w97, w101, w102, w103, w124, w130, w133, w135, w157, w370, w374, w510, A7, A48
\f@shape *308*, r279, r285, r293, r332, r334, r477, r495, r530, s402, s406, s522, s526, s533, s538, s541, s544, s582, v272, w511
\f@size o180, o1185, r88, r288, r294, r333, r478, r517, r551, r593, r594, r597, r598, t119, t121, t134, t154, t169, t172, t175, t180, t187, t194, t206, t209, t215, t221, t238, t239, t242, t247, t313, t320, t339, t341, t356, t407, t409, t411, t427, t428, t433, t447, t459, t464, t476, t484, t489, t515, t529, w406, w415, A123, A840
\f@user@size . . . t427, t432, t476, t489
\famdef@ult w276
\filec@ntents P924, P927, P930, P941, P962, P1056, P1058, P1061, P1072, P1093, P1186, P1208, P1296, P1511
\filec@ntents@checkdir P947, P949, P963, P1078, P1080, P1094, P1193
\filec@ntents@force P943, P1074, P1189
\filec@ntents@noheader P945, P1076, P1191
\filec@ntents@nosearch P946, P1077, P1192
\filec@ntents@opt . . . P927, P930, P932, P1058, P1061, P1063, P1188
\filec@ntents@OPTION *662*
\filec@ntents@overwrite P944, P1075, P1190
\filec@ntents@where . P948, P950, P975, P1079, P1081, P1106, P1194
\filename@area a246, a252, a259, a265, a272, a278, a285, n503, n518, n528, n560, n561, n565, n590, n593, n610, n622, n624, P701
\filename@base a295, a302, a315, n503, n518, n528, n560, n562, n565, n590, n593, n617, n622, P702
\filename@dot a313, a319
\filename@dots a297, a299, a304
\filename@ext *688*, a294, a301, a311, a313, n504, n519, n529, n556, n557, n560, n563, n565, n586, n587, n590, n593, n618, P703
\filename@parse a110, a242, n501, n516, n526, n555, n585, n615, P700
\filename@path . . a247, a248, a253, a260, a261, a266, a273, a274, a279
\filename@simple a250, a263, a276, a286, a290, a292, a307, a309
\finph@nt E104, E106, E110, E111, E119, E120
\finsm@sh E134, E136, E140, E141, E145, E146
\fix@penalty z101
\fixed@sfcnt t519, t520, t521
\fl@trace S240, S267, S323, S351, S358, S379, S426, S472, S525, S540, S541, S542, S543, S554, S555, S556, S557, S558, S568, S781, S800, S819, S837, S839, S978, S982, S994, S995, S996, S997, S1003, S1006, S1014, S1018, S1029, S1034, S1039, S1040, S1041, S1042, S1049, S1052, S1060, S1071, S1077, S1082, S1087, S1093, S1094, S1099, S1104, S1105, S1106, S1114, S1118, S1123, S1127, S1132, S1143, S1144, S1146, S1164, S1173, S1179, S1188, S1191, S1197, S1207, S1211, S1221, S1227, S1233, S1239, S1246, S1248, S1254, S1259, S1261, S1263, S1271, S1276, S1282, S1287, S1293, S1307, S1308, S1311, S1332, S1341, S1347, S1356,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

S1359, S1366, S1376, S1380, S1392, S1398, S1403, S1408, S1412, S1416, S1417, S1424, S1429, S1433, S1440, S1449, S1453, S1457, S1458, S1462, S1463, S1473, S1479, S1485, S1491, S1495, S1501, S1503, S1511, S1516, S1521, S1529, S1538, S1543, S1548, S1550, S1555, S1557, S1568, S1574, S1584, S1590, S1594, S1595, S1600, S1601, S1607, S1610, S1611, S1612, S1619, S1620, S1621, S1629, S1634, S1646, S1647, S1654, S1657, S1665, S1669, S1673, S1674, S1678, S1679, S1689, S1695, S1705, S1711, S1715, S1716, S1722, S1723, S1730, S1733, S1734, S1735, S1743, S1744, S1745, S1754, S1759, S1772, S1774, S1781, S1784, S1793, S1797, S1801, S1802, S1806, S1807, S1859, S1864, S1870, S1880, S1887, S1897, S1993, S2006, S2007, S2011, S2014, S2016, S2019, S2022, S2024, S2065, S2072, S2077, S2083, S2088, S2092, S2098, S2106, S2108, S2115, S2120, S2125, S2127, S2133, S2135, S2142, S2171, S2173, S2188, S2190, S2204, S2229, S2233, S2238, S2250, S2267, S2272, S2280

`\fl@tracemessage` S1897

`\fl@traceval` S1897

`\float@count` b51, b52, b53, b62, b188, b205, b210, b212, b213, b222, b230

`\fmtversion@topatch` U490, U492, U504, U505, U517, U525

`\font@info` t99, t319, t388, t393

`\font@name` o179, o182, r86, r194, r196, r325, r340, r516, r550, t84, t88, t90, t105, t120, t123, t126, t284, t285, t286, t287, t288, t293

`\font@submax` t395, t424, t425, D43, D45, D83, D85, U231, U233, U242

`\fps@dbl` L34

`\frozen@everydisplay` r315, r321

`\frozen@everymath` r315, r319

`\g@addto@macro` . . . *123*, f813, s613, P856, P872, P873, R276, R282, R323

`\G@refundefinedfalse` *456*, C5

`\G@refundefinedtrue` C3, C12, N21, N44

`\gen@sfcnt` t456, t457, t458

`\genb@sfcnt` t461, t462, t463

`\genb@x` t464, t466

`\genb@y` t466

`\get@cdp` v388, v396, v429

`\get@external@font` t83, t96, t490

`\getanddefine@fonts` . . r563, r581, t274, v59, v87, v132, v148, v179, v210, v295, v359, v393, v395, v412, v535, v536, v568, v569, v1011, v1012

`\glb@currsize` n43, n112, n170, r312, t171, t206, t210, t216, t239

`\glb@settings` . . r313, t171, t218, t249

`\group@elt` v35, v293, v330, v331, v352, v356, v1043

`\group@list` v297, v337, v350, v355, v356, v385, v611, v659, v720, v805, v808, v859, v862, v928, v931, v998, v1049

`\h@false` E81

`\h@true` E82, E83

`\hb@xt@` b449, f29, o422, E197, E362, E424, E439, E451, E478, E508, G44, G65, G83, G186, G428, G432, G433, G434, H37, K203, K226, K233, S630, S640, S689, S699, S1843, S2223, S2224, S2228, S2255, S2256, S2262, I31, I43, I62, I74, I105, I117, I265, I299, I302, I305, I307, I314, I373, I452, I588, I598, I741, I797

`\hexnumber@` v632, v640, v675, v683, v704, v714, v740, v748, v756, v765, v768, v777, v778, v817, v825, v871, v879, v898, v899, v909, v910, v915, v941, v949, v954, v956, w495

`\hgl@` b404, b405

`\hmode@bgroup` o67, o75, o327, o356, o390, o396, o424, o435, o442, o473, o480, o483, o485, o493, o509, o724, o754, o760, o792, o799, o827, o830, o876, o1219, z7, A541, A548

`\hmode@start@before@group` o68, o151, o153, o159, o184

`\if@afterindent` K124, K131

`\if@compatibility` P2, P499

`\if@endpe` D203, D210, D262, D277, F138

`\if@eqnsw` E344, E409

`\if@fcolmade` S95, S264, S394, S403, S441, S451, S779, S799, S817, S846, S926, S2170, S2187, S2237, S2277

`\if@filesw` n7, n38, n107, n165, n298, n310, n331, n345, n357, n364, n376, D20, D49, D75, D89, K153, N4, N8, N19, N28, N36, N43, P973, P992, P1104, P1123, R285

`\if@firstamp` H251

`\if@firstcolumn` S95, S246, S279, S396, S444, S1815, S2201, S2246

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

\if@font@series@context *381*, w359, w378, w386
\if@forced@series s385, w78
\if@ignore D4, D205, D212, D263, D278
\if@includeinrelease c68, c71, c98, f767
\if@inlabel F28, F65, F102, F160, F183, S161, S188
\if@insert S95, S1057, S1169, S1203, S1337, S1372, S1446, S1535, S1662, S1790
\if@minipage . l239, l257, l276, l311, D404, D426, F149, G304, H79, L20
\if@mparswitch S95, S1817
\if@multiplelabels C31
\if@negarg I157, I198, I212, I273
\if@newlist D446, F29, F33, F69, F78, F106, F166, S599, S644, S657, S703
\if@nmbrlist F33, F201
\if@no@font@opt u16, u110, u129
\if@nobreak l120, l147, l278, l313, n204, n216, F167, F192, G267, G288, K47, K128, L180, L349, O29, O37, S165, S192, S335, S1148, S1314
\if@noitemarg F32, F199
\if@noparitem F30, F157
\if@noparlist F31, F114
\if@noskipsec ... l147, F58, G268, G289, K38, K40, K97, L350, S155, S182
\if@ovb I427, I495, I526, I551, I562, I575
\if@ovhline I459, I590
\if@ovl I427, I493, I522, I547, I592, I601
\if@ovr I427, I492, I521, I546, I589, I599
\if@ovt I427, I494, I525, I550, I567, I579
\if@ovvline I459, I565
\if@partsw n7, n302, n349
\if@pboxsw G259, G362
\if@reversemargin S101, S1820
\if@reversemarginpar S95
\if@rjfield H19, H33
\if@specialpage S95, S606, S664
\if@tempswa a78, a79, a80, b270, h9, n308, n355, o1502, r99, r658, v318, v373, v437, v518, v1042, D51, D91, D411, D432, N52, P1004, P1135, P1224, Q274, Q275, S990, S1026, S1626, S1751
\if@test S12, S13, S887, S906, S946, S968, S1032, S1116, S1125, S1274, S1285, S1427, S1514, S1632, S1757
\if@twocolumn n28, n97, n154, L32, L210, L235, S95, S139, S267, S278, S395, S443, S467, S781, S837, S1814, S2172, S2189
\if@twoside S95, S138, S609, S667
\ifdt@p E190, E192
\IfFileExists@ n409, n410, n445
\ifG@refundefined C3, C4, C5
\ifh@ E76, E114, E123
\ifin@ o1518, o1521, u50, u52, v1, v22, v282, v384, v386, v447, v460, v530, v532, v560, v612, v626, v660, v672, v721, v737, v806, v809, v830, v860, v863, v926, v929, v932, v999, v1001, v1030, w145, w149, w153, P220, P238, P410, P422
\ifmath@fonts r204, t176
\ifmaybe@ic z82, z91
\ifnot@nil t297, t314, t335
\iftc@forced A780, A791, A1059
\ifv@ E75, E113, E122
\in@ *305*, o1516, o1519, u49, u51, v1, v21, v281, v383, v385, v443, v456, v529, v531, v558, v610, v621, v658, v669, v719, v733, v804, v807, v827, v858, v861, v923, v927, v930, v997, v1000, v1028, w143, w147, w151, P219, P236, P407, P421
\in@@ s432, s433, s435, s436, v5, v6, v7, v9
\in@false v10
\in@true v12
\init@restore@glb@settings t219, t222, t224
\init@restore@version v62, v91, v108, v123, v124
\init@series@setup s614, s619, w141, w182
\input@path *10*, *680*, *688*, *693*, a109, a131, a133, a139, a141, a147, a149, a154, a156, a166, a233, n413, n427, n450, n465
\insc@unt b37, b51, b52, b53, b62, b90, b91, b92, b94, b247, b248, b249, b250, b251, b252, b263, b264, b265, b266, b267, b271, b273, b292, b293, b294, b295, b296, b297, S61
\install@mathalphabet r558, r575, r582, v301, v304, v390, v391, v488, v540, v543, v550, v565, v566, v573, v1013, v1015
\is@range t330, t331
\kernel@ifnextchar c76, f81, f100, f150, f678, f693, P344
\kernel@make@fragile f375, f794, f795, f796, f797, f798, f799, f800, f801, f802, f803, f804, f805, f806, f807, f808, f809, l24, l25, l26,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

l27, l28, o172, o173, D380, D381, D382, E90, E91, E92, E93, E166, E167, E168, H160, H161, H162, K23, K24, K25, K26, K27, O43, O44, I823, I824, I825, I826, I827, I828, I829, I830, I831, I832, I833, I834
\l@ngrel@x f74, f75, f76, f120, f167
\l@nohyphenation .. D408, D548, U216
\last@fontshape r491, r509, r526, r543
\leavevmode@ifvmode l461, l462, l470, x624, x626, x628, x630, E115, E141, E206, E226, E227
\load@onefile@withoptions *656*, P705, P730, P835
\load@onefilewithoptions *659*, P652, P785, P1311
\lower@bound t340, t341, t352
\ltx@sh@ft b441, o390, o397, o473, o481, o754, o761
\m@ne *155*, b39
\m@th b421, b433, m13, x337, x459, x461, x462, x465, x506, x530, x533, x537, x540, x547, x550, x557, x560, x642, E68, E71, E106, E136, E154, E156, E172, E191, E353, E439, E451, E478, E488, G259, G388, H181, K199, K222, L376, L385, L392, L413, L420
\makeph@nt E101, E103
\makesm@sh E131, E133
\mandatory@arg t368, t455, t459, t464, t471, t473, t478, t480, t485, t487, t500, t516, t523, t525
\math@bgroup *345*, r589, t260, t266, v53, v81, v142, v164, v173, v204, z130, z131, z138
\math@egroup r589, t264, t265, z131, z132, z139
\math@fonts r559, r564, t186, t290, v60, v89, v149, v180, v212
\math@fontsfalse m7, o302, o329, o360, o1221, r77, r206, r216, r239, A124, A584, A841
\math@fontstrue r204, r601
\math@version r7, r307, r563, r567, r569, r570, r572, t184, v56, v59, v64, v65, v69, v84, v88, v93, v94, v98, v111, v112, v113, v126, v127, v128, v145, v148, v152, v154, v156, v160, v176, v179, v183, v185, v187, v191, v207, v211, v215, v217, v219, v223, w467
\mathchar@type . v704, v714, v765, v768, v777, v793, v898, v909, v972
\mathph@nt E99, E105
\mathsm@sh E129, E135
\maybe@ic z63, z64, z83
\maybe@ic@ z83
\maybe@icfalse z97
\maybe@ictrue z87
\maybe@load@fontshape *220*, o71, s401, s423, s535, s590, w99
\mb@b G55, G66, G74, G84
\mb@l G55, G59, G65, G74, G78, G83, I137, I141
\mb@r G55, G59, G65, G74, G78, G83, I137, I141
\mb@t G56, G63, G75, G82
\md@def@ult w190
\mddef@ult w133, w195, w224, w225, w226, w275
\mddefault@previous w221, w223, x107, x117
\mdseries@.. ..... *369*, *374*
\mdseries@rm w63, w173, w224, w229, w293
\mdseries@sf w63, w174, w225, w230, w294
\mdseries@tt w63, w175, w226, w231, w295
\merge@font@series *303*, *303*, s387, s388, s587
\merge@font@series@ . s390, s395, s588
\merge@font@shape ... s521, s523, s595
\merge@font@shape@ .. s525, s530, s596
\n@space x625, x627, x629, x631, x636, x637, x638, x639, x642
\new@command f77, f78, f131, f165, f184, f239
\new@environment f146, f147, f159
\new@fontshape u2, u4, u22, u24
\new@mathalphabet ... v441, v462, v473
\new@mathgroup b78, b80, b98, b100, d27, r14, v321
\new@mathversion v20, v278, v280
\new@symbolfont v322, v354
\newmathalphabet@ u14
\newmathalphabet@@ u109
\newmathalphabet@@@ u15, u109
\nfss@catcodes *283*, r19, r120, r358, r359, r366, r413, x40, x45, x135, S3
\nfss@text o315, o317, w498, z5, z122, C13
\no@alphabet@error r4, v300, v302, v478, v479, v493, v502, v588, v589
\noaccents@ r604, x129
\non@alpherr r583, r585, v72, v101, v117, v163, v194, v226, v1050

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\not@base w533, w537, w538, w539, w540, w541, w542, w543, w544, w545, w546, w547
\not@math@alphabet s450, s455, s460, s551, s555, s558, s561, s564, s567, s570, s573, w5, w8, w11, w14, w17, w20, w23, w26, w29, w199, w219, w238, w243, w248, w280, w291, w302, w307, w312, w325, w328, w331, w334, w337, w447
\now@and@everyjob . . . d207, d213, o994
\o@lign b435, o390, o397, o473, o481, o754, o761
\on@line g196, g416, j8, j15, j165, w79, w82, D166, D221, D238, D271, G131, P713, P824
\operator@font x643, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E22, E23, E24, E25, E26, E27, E28, E29, E30, E31, E32, E33, E34, E37, E40
\optional@arg t369, t448, t450, t522, t525
\outer@nobreak L181, L251, L255, L342
\p@ b309
\p@enum *509*
\p@equation E351, E487
\p@selectfont t117
\par@deathcycles . . F56, F77, F79, F80
\patch@level *34*, c1, c36, c41, c43, c45, c48, c56, U493, U505, U507
\ph@nt E81, E82, E83, E97
\pickup@font o181, r195, r324, r518, r552, t122, t285, t287, t289
\pictur@ I21
\pkgcls@arg P1322, P1445
\pkgcls@candidate . . P1309, P1324, P1400, P1404, P1408, P1476, P1479
\pkgcls@debug P1299, P1315, P1316, P1317, P1318, P1319, P1376, P1377, P1378, P1379, P1388, P1393, P1411, P1420, P1435, P1469, P1470, P1471
\pkgcls@innerdate P1304, P1349, P1352, P1358, P1497
\pkgcls@mindate *672*, P1329, P1338, P1354, P1359
\pkgcls@name P1321, P1364
\pkgcls@parse@date@arg . P1323, P1334
\pkgcls@parse@date@arg@ P1340, P1343
\pkgcls@parse@date@arg@version P1350, P1371
\pkgcls@releasedate P1309, P1405, P1409, P1480
\pkgcls@rollbackdate@error P1401, P1459, P1477
\pkgcls@show@selection P1428, P1433, P1483, P1488
\pkgcls@targetdate . . . *672*, P1304, P1336, P1344, P1347, P1348, P1352, P1360, P1361, P1374, P1382, P1397, P1399, P1429, P1440, P1442, P1467, P1473, P1475
\pkgcls@targetlabel *672*, P1304, P1337, P1357, P1372, P1384, P1416, P1449, P1487, P1490
\pkgcls@use@this@release . P1385, P1402, P1404, P1417, P1427, P1479
\pr@@@s E246, E254
\pr@@@t E249, E255
\pr@m@s E243, E244
\preload@sizes u11, u94
\prepare@family@series@update *378*, w76, w179, w239, w244, w249, w303, w308, w313
\prim@s E240, E242, E254
\prime@s E241
\process@table . n42, n111, n169, v232
\protected *146*
\protected@edef *228*, *362*, *809*, f253, p158, w396, C43, C47, C55, C59, C66, G356, K60, L443, U470, U479, U484, U485
\protected@file@percent *464*, *465*, *465*, *597*, D111, D118, D133, D141, D142, K165, K172
\protected@wlog P290, P292, P306, P320
\protected@write n203, n208, C33, K181, M14, M31
\protected@xdef f253, K11, L429, L453, L469, P282, P301
\provide@command f178, f179
\ps@empty O10, U91
\ps@plain O13
\q@curr@file P924, P964, P966, P971, P999, P1095, P1097, P1102, P1130
\quote@@name n400, n442
\quote@name n400, n428, n430, n439, n441, P1095
\r@@t E66
\reenable@package@load *683*, *696*, *719*, Q413, R385
\reinstall@nfss@defs s553, s606, s612, s614, s618, s619, s622
\rem@pt r300
\remove@angles t301, t324
\remove@nil v36

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\remove@star t301, t307
\remove@tlig o984, o986, o988, o1013, o1018, o1054, o1056, o1058
\remove@to@nnil r299, t301, t327, t440
\renew@command .. f124, f125, f185, f193
\renew@environment f152, f153
\requested@test@context *381*, w353, w370, w374
\reserved@@b *358*
\reserved@a . *381*, a121, a125, a126, a195, a196, a199, a217, a221, a243, a250, a253, a255, a256, a263, a266, a268, a269, a276, a279, a281, a331, a332, a333, b193, c13, c19, c34, e28, e33, e70, e76, f117, f120, f133, f134, f135, f137, f184, f185, f186, f192, f193, f194, f195, f198, f218, f226, f230, f288, f292, f326, f330, f354, f358, f469, f477, f478, f526, f529, f544, f546, f549, f558, f579, f586, f608, f611, f675, f684, i33, i37, j185, l409, l412, n213, n214, n236, n245, n254, n306, n353, n414, n416, n421, n423, n425, n431, n435, n451, n453, n458, n460, n463, n469, n473, n502, n505, n506, n508, n517, n520, n527, n530, n552, n553, n554, n558, n566, n583, n584, n588, n594, n616, n620, n624, o81, o82, o86, o89, o97, o107, o110, o119, o138, o143, o978, o982, o1032, o1041, r30, r31, r32, r41, r44, r47, r63, r66, r69, r105, r108, r110, r147, r151, r360, r363, r490, r491, r506, r509, r514, r525, r526, r539, r543, r548, r575, r578, r579, r587, s397, s398, s402, s403, s418, s419, s531, s532, s536, s537, t150, t152, t154, t164, t166, t169, t298, t299, t312, t313, u53, u57, v388, v397, v399, v443, v446, v456, v459, v557, v559, v621, v625, v669, v671, v732, v735, v827, v829, v923, v925, v1027, v1029, v1045, v1047, v1048, v1053, w37, w38, w109, w110, w366, w367, z47, z48, z53, z54, z65, z68, z88, z95, D109, D110, D164, D165, D170, D219, D220, D224, D236, D237, D241, D269, D270, D274, D282, D283, E404, E405, E406, E407, E409, G57, G58, G61, G76, G77, G80, G126, G132, H241, H245, H250, H269, H360, H361, L29, L30, L32, L33, L63, L67, L72, L74, L76, L78, L83, L84, L132, L136, L142, L145, L148, L151, P214, P222, P226, P232, P240, P244, P352, P354, P355, P356, P360, P479, P483, P489, P493, P555, P556, P559, P600, P604, P616, P617, P619, P628, P632, P644, P645, P646, P648, P657, P660, P661, P663, P733, P779, P792, P833, P935, P936, P938, P980, P981, P983, P987, P1066, P1067, P1069, P1111, P1112, P1114, P1118, P1326, P1331, P1383, P1384, P1415, P1416, P1486, P1487, P1511, P1513, Q150, Q158, Q164, Q165, S37, S46, S48, S50, S877, S897, S1963, S1965, S1966, S2055, S2057, S2063, S2066, U159, U176, U177, U178, U185, U186, U187, U389, U392, U423, U429, U430, U441, U442, U443, U450, U451, U452, U465, U466, U470, U471, U474, U475, U479, U480, U506, U509, U510, U527, I199, I201, I205, I478, I479, I506, I507, I535, I536
\reserved@b a122, a123, e29, e35, f109, f111, f118, f135, f136, f227, f228, f230, f289, f290, f292, f327, f328, f330, f355, f356, f358, f527, f529, f545, f549, f583, f586, f676, f686, i33, i34, i37, l410, l411, l418, n304, n306, n351, n353, n426, n428, n430, n464, n466, n468, n552, n553, n554, n619, n625, o90, o97, o112, o119, o981, o982, o1041, o1050, o1052, r31, r32, r38, r95, r97, r150, r151, r576, r587, s417, s418, s420, u47, u54, u71, u73, v314, v316, v369, v371, v396, v397, v398, v433, v435, v514, v516, v561, v562, v563, v570, v730, v734, v736, w116, w121, w125, w126, w133, w134, z52, z53, z66, z68, z95, z96, H246, H248, H250, L43, L44, L112, L113, P215, P216, P217, P219, P234, P237, P352, P608, P614, P617, P636, P642, P646, P657, P664, P982, P983, P986, P987, P1022, P1023, P1025, P1051, P1113, P1114, P1117, P1118, P1153, P1154, P1156, P1182, P1242, P1243, P1245, P1272, S786, S789, S803, S806, S823, S826, U162, U164, U168,

File Key: a=`ltdirchk.dtx`, b=`ltplain.dtx`, c=`ltvers.dtx`, d=`ltluatex.dtx`, e=`ltexpl.dtx`, f=`ltdefns.dtx`, g=`lthooks.dtx`, h=`ltalloc.dtx`, i=`ltcntrl.dtx`, j=`lterror.dtx`, k=`ltpar.dtx`, l=`ltspace.dtx`, m=`ltlogos.dtx`, n=`ltfiles.dtx`, o=`ltoutenc.dtx`, p=`ltcounts.dtx`, q=`ltlength.dtx`, r=`ltfssbas.dtx`, s=`ltfssaxes.dtx`, t=`ltfsstrc.dtx`, u=`ltfsscmp.dtx`, v=`ltfssdcl.dtx`, w=`ltfssini.dtx`, x=`fontdef.dtx`, y=`preload.dtx`, z=`ltfntcmd.dtx`, A=`lttextcomp.dtx`, B=`ltpageno.dtx`, C=`ltxref.dtx`, D=`ltmiscen.dtx`, E=`ltmath.dtx`, F=`ltlists.dtx`, G=`ltboxes.dtx`, H=`lttab.dtx`, I=`ltpictur.dtx`, J=`ltthm.dtx`, K=`ltsect.dtx`, L=`ltfloat.dtx`, M=`ltidxglo.dtx`, N=`ltbibl.dtx`, O=`ltpage.dtx`, P=`ltclass.dtx`, Q=`ltfilehook.dtx`, R=`ltshipout.dtx`, S=`ltoutput.dtx`, T=`lthyphen.dtx`, U=`ltfinal.dtx`

U395, U397, U401, U466, U475, U527
\reserved@c . a123, a128, f681, f684, f686, f689, n609, r96, r97, r577, r580, u48, u55, u61, u68, v33, v37, v315, v316, v370, v371, v434, v435, v515, v516, v538, v547, v562, v576, v817, v834, v843, v871, v882, v940, v953, v955, w118, w121, w131, w132, w135, w136, z67, z69, z76, P658, P660, P661, P965, P970, P971, P991, P999, P1005, P1027, P1034, P1096, P1101, P1102, P1122, P1130, P1136, P1158, P1165, P1213, P1214, P1215, P1225, P1247, P1254, P1282, U166, U171, U179, U389, U399, U420, U421, U422, U424, U425, U426, U427, U428, U436, U444, U529
\reserved@d a126, a129, e27, e32, f674, f683, n607, n609, u61, u68, u70, u74, v825, v834, v843, v879, v882, v948, v953, v957, w123, w124, w129, w130, U530
\reserved@e l57, l59, l69, l71, l100, l107, l115, u39, u45, u70, u73, u74, v34, v39, U531
\reserved@f l58, l59, l70, l71, l115, o1503, o1504, o1505, o1506, o1508, o1515, r190, r192, r198, r199, t336, t347, t351, t355, t361, t364, t403, t440, t443, u27, u38, u45, u71, u73, U532
\reset@font *373*, w154, w503, w527, C13, G353, L175, L347, L438, N20, O14, S616, S675
\restglb@settings t222, t232
\restore@mathversion v107, v110, v125, v133
\restore@protect f253
\rlh@ x464, x465
\rm@def@ult w190
\rmdef@ult . . *375*, *381*, w191, w210, w229, w271, w282, w293, A27, A105
\robust@command@act *86*, *88*, *89*, f410, f411, f413, f479, f503
\robust@command@act@chk@args *86*, *87*, f435, f456
\robust@command@act@do . *87*, f421, f453
\robust@command@act@end *86*, f418, f419, f431, f434, f454
\robust@command@act@loop *87*, f415, f421, f451
\robust@command@act@loop@aux f421, f452
\robust@command@chk@safe *83*, *92*, f303, f414, f435, f455, f560, f572
\s@fct@ t380, t444
\s@fct@alias t506
\s@fct@fixed t519
\s@fct@gen t456
\s@fct@genb t461
\s@fct@sgen t456
\s@fct@sgenb t461
\s@fct@sub t468
\s@fct@subf t511
\saved@space@catcode . . . U253, U322
\scan@@fontshape u7, u40, u43
\scan@fontshape u6, u26, u37
\scriptfont@name t287, t292
\select@group r560, r579, v48, v268, v305, v443, v496, v505, v543, v575
\series@change@debug w72, w79, w82, w93, w96, w100, w112, w120, w125, w131, w134, w136
\series@check@toks . . s433, s435, s442
\series@drop@one@m . . s439, s443, s593
\series@maybe@drop@one@m r31, s426, s428, s592, w103, w122, w128, w194, w195
\series@maybe@drop@one@m@x s429, s431
\seriesdefault@kernel *389*, w155, w586
\set@@mathdelimiter v880, v914
\set@color G88
\set@curr@file *656*, *656*, *691*, *691*, *693*, *693*, *694*, n227, n236, n263, n271, n390, n408, P692, P963, P1094, Q258
\set@display@protect f8, f16, f251, j7, j14, j34, j61, P293
\set@fontsize . . r288, r290, t119, t132
\set@mathaccent v630, v638, v673, v681, v699
\set@mathchar v754, v764
\set@mathdelimiter . . v831, v840, v892
\set@mathradical v276, v950
\set@mathsymbol v738, v746, v767
\set@simple@size@args t302, t315, t322, t343, t357
\set@size@funct@args t305, t307, t365
\set@size@funct@args@ t365
\set@target@series s399, s404, s407, s410, s424, s591
\set@typeset@protect f251, f270, H197, H235, R45, S603, S605, S661, S663
\SetMathAlphabet@ . . . v450, v519, v528
\SetSymbolFont@ v340, v374, v382
\sf@def@ult w190

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\sf@size m6, o302, r224, r243, r599, t282, t286, A584, L385, L392, L413, L420
\sfdef@ult *375*, w192, w211, w230, w272, w283, w294, A29, A107
\sh@ft b439
\show@kernel@robust@command f566, f626
\sixt@@n a71, b16, b64, b66, b96, b97, b98, d30, r14, v84, v207, v616, v618, v664, v666, v725, v727, v773, v775, v813, v815, v821, v823, v867, v869, v875, v877, v936, v938, v944, v946, L62, L80, L131, L153, S1005, S1051, S1190, S1358, S1592, S1656, S1713, S1783, S2009, S2018, S2074, S2090, S2123, I278, I293, I295
\sixt@@n␣ *155*
\size@update . . . t128, t139, t158, t160
\sizefn@info t306, t308, t316, t344, t358
\skip@ b41, b401, b403, b404, b406, l83, l442, z105, z108
\sp@n H379
\split@name r328, r340, r454, r469, t473, t487
\ssf@size o328, o360, o1220, r225, r244, r600, t282, t288
\string@makeletter f701, P663, P701, P702, P703, Q165
\strip@prefix a111, a228, a323, f228, f290, f328, f356, f698, r557
\strip@pt b443, r216, r222, r223, r224, r225, r238, r242, r300, r599, r600, t134
\sub@sfcnt t468, t469, t470, t497
\subf@sfcnt t511, t512, t513
\subst@correction r85, r91
\subst@fontshape u8, u80
\subst@size t419
\sw@slant z91, z101
\t@st@ic z90, z94
\target@meta@family@value w85, w110, w117, w119
\target@series@value . . *371*, *376*, *377*, w84, w92, w95, w97, w101, w102, w103, w126, w132, w133, w135
\tc@check@accent A130, A180, A181, A182, A183, A184, A185, A186, A187, A188, A189, A190, A191, A192, A193, A194, A195, A847, A923, A925, A927
\tc@check@symbol A130, A214, A215, A216, A217, A218, A219, A220, A221, A222, A223, A224, A225, A226, A227, A228, A229, A230, A231, A232, A233, A234, A235, A236, A237, A238, A239, A240, A241, A242, A243, A244, A245, A246, A247, A248, A249, A250, A251, A252, A253, A254, A255, A256, A257, A258, A259, A260, A261, A262, A264, A265, A266, A267, A268, A269, A270, A271, A272, A273, A274, A275, A276, A277, A278, A279, A280, A281, A282, A283, A284, A287, A288, A289, A290, A291, A292, A293, A847, A917, A919, A921, A929, A931, A933, A935, A937, A939, A941, A943, A945, A947, A949, A951, A953, A955, A957, A959, A961, A963, A965, A967, A969, A971, A973, A975, A977, A979, A981, A983, A985, A987, A989, A991, A993, A995, A997, A999, A1001, A1003, A1005, A1007, A1009, A1011, A1013, A1015, A1017, A1019, A1021, A1023, A1025, A1027, A1029, A1031
\tc@error A131, A827, A848
\tc@errorwarn A21, A97, A99, A736, A737, A738, A739, A786, A787, A788, A821
\tc@fake@euro A118, A286, A835, A916
\tc@forcedfalse A780
\tc@forcedtrue A785
\tc@oldstylesubst A16, A20
\tc@subst A98, A130, A820, A847
\tc@swap@accent A132, A133
\test@font@series@context *381*, w355, w365, w385
\text@command z8, z46
\textfont@name t285, t291
\tf@size . . r223, r243, r598, t282, t284
\thr@@ b16, b484, t58, t208, t214, t227, t234, t241, t246, E362, E507, F232, F243, U70, U78, I287, I288, I290, I291, I329, I355, I388, I411
\toks@ *305*, b41, c81, c85, c88, c93, f815, f816, l408, l409, l414, r148, r152, r154, r157, r221, r226, v6, v7, v291, v295, v301, v304, v309, v355, v356, v358, v359, v389, v391, v395, v412, v415, v474, v486, v487, v488, v534, v536, v542, v550, v554, v566, v569, v572, v580, v582, v1003, v1005, v1007, v1010, v1012, v1015, v1018,

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

v1050, v1051, P383, P384, P386, P387, S2209, S2210, S2211, S2212
\try@load@fontshape ... *304*, r343, r351, r397, r502, s423, t474, v240, v257
\try@simple@size t310, t435
\try@simples t393, t399, t403
\try@size@range t101, t310, t386
\try@size@substitution ... t103, t390
\tryif@simple t401, t402
\tryis@simple t402
\tt@def@ult w190
\ttdef@ult *375*, w193, w212, w231, w273, w284, w295, A31, A109
\tw@ *155*, b16
\two@digits a86, a185, a186, f2, t466, P920, P1011, P1142, P1231
\type@restoreinfo t156, t161
\unclare@... ..... *718*
\undeclare@file@substitution *682*, Q238, R382
\unqu@tefilef@und Q140
\unquote@name . n394, n400, n443, Q287
\unrestored@protected@xdef f253, L434, L458, L474, O25, O49
\update@series@target@value w87, w108, w180
\update@uclc@with@cyrillic o1465, o1493, o1523, o1531
\upper@bound ... t337, t338, t339, t352
\use@mathgroup r566, r584, r586, t253, v63, v92, v456, v558, v561, v1028, v1052
\UTF@four@octets@noexpand U469, U478
\UTF@three@octets@noexpand U468, U477
\UTF@two@octets@noexpand U467, U476
\UTFviii@four@octets U308, U313, U319
\UTFviii@four@octets@@ .. U308, U319
\UTFviii@four@octets@combine . U343
\UTFviii@four@octets@noexpand . U349
\UTFviii@four@octets@string .. U346
\UTFviii@invalid U247, U340
\UTFviii@invalid@err U305, U310, U316
\UTFviii@invalid@err@@ .. U305, U316
\UTFviii@three@octets U307, U312, U318
\UTFviii@three@octets@@ . U307, U318
\UTFviii@three@octets@combine . U342
\UTFviii@three@octets@noexpand U348
\UTFviii@three@octets@string . U345
\UTFviii@two@octets U306, U311, U317
\UTFviii@two@octets@@ .. U306, U317
\UTFviii@two@octets@combine .. U341
\UTFviii@two@octets@noexpand . U347
\UTFviii@two@octets@string ... U344
\UTFviii@undefined@err U304, U309, U315
\UTFviii@undefined@err@@ U304, U315
\v@false E82
\v@true E81, E83
\ver@⟨*file*⟩.⟨*ext*⟩ *655*
\verb@balance@group . D494, D496, D509, D511, D524, D526, D532, D533
\verb@egroup D494, D497, D509, D512, D524, D527, D533, D537
\verb@eol@error .. D534, D546, D556
\verbatim@font D418, D439, D447, D547, D557
\verbatim@nolig@list ... D562, D568
\version@elt v18, v31, v32, v288, v289, v338, v358, v449, v487, v579, v1008
\version@list v16, v21, v32, v281, v289, v343, v364, v383, v454, v499, v529, v584, v1021
\vgl@ b401, b402
\voidb@x b309, b432, q18
\warn@rel@i u5, u25, u29, u81, u85, u90, u95, u119, u140
\wrong@fontshape r347, r484
\x@protect f231, f242, f293, f331, f359, f530, f550
\xe@alloc@ U42, U52
\xe@alloc@intercharclass U21
\xe@ch@ck U43, U47
\z@ *155*, b309
\z@skip b309
\zap@space . n263, n283, w396, N29, P215, P357, P370, P478, P578, P599, P616, P627, P644, P933, P1064
tex commands:
\tex_afterassignment:D R28
\tex_aftergroup:D *706*, R36
\tex_currentgrouplevel:D .. R27, R35
\tex_deadcycles:D R52
\tex_escapechar:D Q204, Q217
\tex_setbox:D R29
\tex_shipout:D R88, R295
\tex_vss:D R258
\TexOrMath p127, p143
\textacutedbl o898, o1128, A226, A631, A880
\textascendercompwordmark o848, A174, A595, A863
\textasciiacute o948, o1089, A227, A632, A904
\textasciibreve o896, o1127, A228, A633, A877

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\textasciicaron o897, o1126, A229, A634, A878
\textasciicircum o286, o555, o1063
\textasciidieresis o936, o1076, A230, A635, A894
\textasciigrave o887, o1055, A231, A636, A875
\textasciimacron o943, o1084, A232, A637, A899
\textasciitilde o287, o556, o1068
\textasteriskcentered o267, o711, o858, o1181, p130, p136, A140, A503, A558, A870
\textbackslash . . . o268, o557, o712, o1062
\textbaht o922, o1131, A235, A641, A1010, A1011
\textbar o269, o558, o713, o1066
\textbardbl o270, o714, o902, o1136, p135, A147, A293, A504, A566, A707, A883
\textbf *370*, z19
\textbigcircle o723, o875, o1198, A236, A642, A962, A963
\textblank o855, o1195, A282, A695, A932, A933
\textborn o888, A237, A314, A643, A968, A969
\textbraceleft o271, o308, o559, o715, o1065
\textbraceright o272, o309, o560, o716, o1067
\textbrokenbar o934, o1074, A148, A567, A892
\textbullet o273, o717, o904, o1145, A141, A505, A559, A885
\textcapitalcompwordmark o847, A173, A594, A862
\textcelsius o905, o1164, A149, A287, A568, A701, A886
\textcent . o930, o1070, A150, A569, A889
\textcentoldstyle *433*, o907, A238, A319, A322, A644, A984, A985
\textcircled *242*, *429*, o279, o283, o300, o301, o724, o876, A175, A176, A570, A583, A597, A767, A1032, A1034
\textcircledP o941, o1166, A239, A645, A1026, A1027
\textcolonmonetary o909, o1157, A266, A675, A986, A987
\textcommaabove o353, o355, o369, o370, o455, o456, o694, o695
\textcommabelow o324, o326, o332, o333, o697, o698, o699, o700, o701, o702, o703, o704, o705, o706, o1218, o1415, o1416, o1417, o1418
\textcompsubstdefault A34, A39, A53, A112, A825
\textcompwordmark *227*, o290, o291, o561, o1133
\textcopyleft o939, A240, A295, A646, A1024, A1025
\textcopyright o283, o317, o937, o1077, A151, A570, A895
\textcurrency o932, o1072, A277, A688, A762, A766, A920, A921
\textdagger o275, o312, o719, o900, o1143, p131, p137, A143, A507, A561, A881
\textdaggerdbl o274, o313, o718, o901, o1144, p132, p138, A142, A506, A560, A882
\textdblhyphen o859, A242, A296, A648, A934, A935
\textdblhyphenchar o895, A241, A297, A647, A980, A981
\textdegree o944, o1085, A152, A571, A900
\textdied o890, A243, A315, A649, A972, A973
\textdiscount o924, o1156, A244, A650, A1014, A1015
\textdiv . . o961, o1110, A153, A572, A914
\textdivorced o889, o1201, A245, A651, A970, A971
\textdollar . . . o255, o307, o435, o562, o792, o856, o1059, A134, A135, A539, A541, A868, A1040, A1042
\textdollaroldstyle *433*, o906, A246, A320, A321, A652, A982, A983
\textdong o918, o1161, A267, A676, A1004, A1005
\textdownarrow o886, o1176, A268, A677, A966, A967
\texteightoldstyle o869, A216, A311, A620, A952, A953
\textellipsis o296, o321, o1146
\textemdash . o256, o407, o563, o779, o1135
\textendash . o257, o408, o564, o780, o1134
\textestimated o925, o1172, A274, A684, A765, A918, A919
\texteuro o959, o1162, A285, A286, A699, A763, A915, A916
\textexclamdown o258, o409, o411, o565, o781, o1069
\textfiveoldstyle o866, A217, A308, A621, A946, A947
\textfloatsep S728, S741, S2104, S2154, S2303

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\textflorin o908, o1124, A276, A687, A887
\textfont t291, E238
\textfouroldstyle o865, A218, A307, A622, A944, A945
\textfraction S1917, S1920, S1944, S1947, S2096, S2297
\textfractionsolidus o860, o1153, A278, A690, A871
\textgravedbl o899, o1129, A233, A638, A879
\textgreater o281, o566, o734, o1061
\textguarani o912, A247, A318, A653, A992, A993
\textheight n24, n25, n93, n94, n150, n151, L257, L258, L261, L287, L301, R295, S78, S225, S226, S274, S399, S447, S474, S645, S704, S763, S815, U89, U90
\texthyphen o260, o414, o568, o783
\texthyphenchar . . o259, o413, o567, o782
\textinterrobang o916, o1152, A284, A697, A1000, A1001
\textinterrobangdown o917, o1202, A283, A696, A1002, A1003
\textit z21
\textlangle o871, o1193, A264, A672, A956, A957
\textlbrackdbl o883, A214, A316, A617, A873
\textleaf o891, A248, A300, A654, A974, A975
\textleftarrow o853, o1173, A269, A678, A928, A929
\textlegacyasteriskcentered A515, A710
\textlegacybardbl A515, A711
\textlegacybullet A515, A712
\textlegacydagger A515, A714
\textlegacydaggerdbl A515, A713
\textlegacyparagraph A515, A715
\textlegacyperiodcentered . . A515, A716
\textlegacysection A515, A717
\textless o280, o569, o733, o1060
\textlira o914, o1158, A270, A679, A996, A997
\textlnot . o940, o1082, A154, A573, A897
\textlquill o928, o1154, A249, A655, A1020, A1021
\textmarried o892, o1200, A250, A656, A976, A977
\textmd z19
\textmho o874, o1171, A251, A657, A960, A961
\textminus o872, o1177, A281, A693, A872
\textmu . . . o949, o1090, A280, A692, A905
\textmusicalnote o893, o1199, A252, A658, A978, A979
\textnaira o911, o1159, A253, A659, A990, A991
\textnineoldstyle o870, A219, A312, A623, A954, A955
\textnormal z15
\textnumero o923, o1165, A275, A685, A1012, A1013
\textogonekcentered o484, o692, o693
\textohm o882, o1170, A279, A691, A764, A917
\textonehalf o957, o1099, A155, A574, A911
\textoneoldstyle o862, A220, A304, A624, A938, A939
\textonequarter o956, o1098, A156, A575, A910
\textonesuperior o953, o1093, A157, A288, A576, A702, A908
\textopenbullet o926, o1197, A254, A660, A1016, A1017
\textordfeminine o305, o938, o1078, A158, A577, A896
\textordmasculine o306, o954, o1094, A159, A578, A909
\TextOrMath . . . p130, p131, p132, p133, p134, p135, p136, p137, p138, p150
\textparagraph o276, o310, o720, o950, o1091, p134, A144, A508, A562, A906
\textperiodcentered o277, o721, o951, o1092, A145, A509, A563, A907
\textpertenthousand *431*, o489, o920, o1148, A262, A263, A669, A1006, A1007, A1044
\textperthousand o487, o903, o1147, A138, A139, A555, A884, A1043
\textpeso o913, o1163, A255, A661, A994, A995
\textpilcrow o921, A256, A313, A662, A1008, A1009
\textpm . . . o945, o1086, A160, A579, A901
\textquestiondown o261, o410, o412, o570, o784, o1101
\textquotedbl o573, o1057
\textquotedblleft o262, o415, o571, o785, o1140
\textquotedblright o263, o416, o572, o786, o1141
\textquoteleft o264, o417, o574, o787, o1137
\textquoteright o265, o418, o575, o788, o1138
\textquotesingle o857, o1053, A161, A580, A869

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\textquotestraightbase o849, A162, A298, A581, A864
\textquotestraightdblbase o850, A163, A299, A582, A865
\textrangle o873, o1194, A265, A673, A958, A959
\textrbrackdbl o884, A215, A317, A618, A874
\textrecipe o915, o1167, A257, A663, A998, A999
\textreferencemark o952, o1151, A258, A664, A1028, A1029
\textregistered o300, o301, o942, o1083, A164, A583, A898
\textrightarrow o854, o1175, A271, A680, A930, A931
\textrm z15
\textrquill o929, o1155, A259, A665, A1022, A1023
\textsc z21
\textsection o278, o311, o576, o722, o935, o1075, p133, A146, A510, A511, A564, A893
\textservicemark o927, o1168, A260, A666, A1018, A1019
\textsevenoldstyle o868, A221, A310, A625, A950, A951
\textsf z15
\textsixoldstyle o867, A222, A309, A626, A948, A949
\textsl z21
\textssc s459, z31, z39
textssc z25
\textsterling . o266, o319, o442, o577, o799, o931, o1071, A136, A137, A540, A548, A890, A1039, A1041
\textstyle m15, x469, E63
\textsubscript L395
\textsuperscript o303, o305, o306, A577, A578, A589, L378
\textsurd o955, o1192, A261, A667, A1030, A1031
\textsw s454, z30, z38
textsw z25
\TextSymbolUnavailable o3, o752
\textthreeoldstyle o864, A223, A306, A627, A942, A943
\textthreequarters o958, o1100, A166, A586, A912
\textthreequartersemdash o852, A165, A289, A302, A585, A703, A867
\textthreesuperior o947, o1088, A167, A290, A587, A704, A903
\texttildelow o894, o1130, A234, A639, A876
\texttimes o960, o1104, A168, A588, A913
\texttrademark o303, o919, o1169, A169, A589, A888
\texttt z15
\texttwelveudash o851, A170, A291, A301, A590, A705, A866
\texttwooldstyle o863, A224, A305, A628, A940, A941
\texttwosuperior o946, o1087, A171, A292, A591, A706, A902
\textulc s449, z28, z29, z35, z37
textulc z25
\textunderscore .. o288, o315, o578, o1064
\textup *307*, z21
\textuparrow o885, o1174, A272, A681, A964, A965
\textvisiblespace o292, o579, o1196
\textwidth n26, n95, n152, G327, L266, S79, S144, S201, S218, S630, S640, S689, S699, S2223, S2255, U90
\textwon o910, o1160, A273, A682, A988, A989
\textyen .. o933, o1073, A172, A592, A891
\textzerooldstyle o861, A225, A303, A629, A936, A937
\TH o532, o1106, U483
\th o580, o1112, U483
\thanks *588*
\thanks K10, K26
thebibliography (environment) *625*
\theenum *509*
\theequation E339, E351, E426, E487
\thefootnote L372, L453, L458, L478
\thempfn G329, L429, L434, L469, L474, L477
\thempfootnote G329, L374
\thepage *714*, n210, B6, C14, C34, K164, K171, K177, M15, M32, N23, O14, S244, S275, S1827
\Theta x299
\theta x275
\thetotalpages *703*, *714*, R274, R355
\thicklines I124
\thickmuskip x646, E215, E217, E230
\thickspace E201
\thinlines I124, I816, I833
\thinmuskip . x644, E207, E209, E225, E231
\thinspace l460, l466, l467, E176, E201, E238
\thispagestyle O6
\tilde x519
\time a179, a183
\times x396
\title *588*

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\title K6, K7, K21, K23, K31
tl commands:
\c_empty_tl g768
\c_novalue_tl *124*
\c_space_tl Q480, Q483
\tl_const:Nn R235, R240
\tl_count:N Q473
\tl_gclear:N *144*, *146*, g82, g481
\tl_gput_left:Nn g425
\tl_gput_right:Nn *136*, g309, g427, g513, g765
\tl_gremove_all:Nn *131*
\tl_gremove_once:Nn *131*, *131*, g37, g37, g314
\tl_gset:Nn g69, g244, g247, g349, g355, g361, g365, g367, g421, g764, g920, Q51
\tl_gset_eq:NN g927
\tl_if_empty:N *148*
\tl_if_empty:NTF g78, g124, g126, g648, g761, g763
\tl_if_empty:nTF g98, g111, g114, g237, g277, Q40, Q46, Q58, Q106, Q112, Q114, Q116, Q352
\tl_if_empty_p:N g835, R293
\tl_if_empty_p:n g260
\tl_if_exist:N *146*
\tl_if_exist:NTF g76, g771, g811, g819, g843
\tl_if_in:NnTF g319
\tl_if_novalue:nTF g89
\tl_log:n g37, g39, g619
\tl_new:N g25, g26, g27, g29, g30, g34, g49, g52, g61, g327, g443, g444, g445, Q8, Q9, Q10, Q11, Q59, R32, R129
\tl_set:Nn g450, g455, g470, g471, g476, g477, g492, g499, g500, g544, g551, g821, Q80, Q81, Q82, Q83, R26, R134, R160, R185, R212
\tl_set_eq:NN g480, g503
\tl_show:n g37, g38, g624, g680
\tl_to_str:n g138, g196, g324, g643, g699, Q507
\tl_trim_spaces:n ... g167, Q41, Q42
\tl_trim_spaces_apply:nN g92
\tl_use:N g610
\tmspace *486*, E201
\to x440, x442
\today a184, a188, a196, a199, K33
token commands:
\token_if_eq_meaning:NNTF Q390, Q402
\token_to_str:N g135, Q232
\toks b31, b63, b95, d36, v485, v486, v496, v505, U533
\toksdef b46, b63, b95, d222
\tokszero d222
\tolerance b315, r618, r663, r678, O58, O66
\top x320
\topfigrule S727, S2325
\topfraction L273, S2291
\topmargin S71, S624, S683
\topmark S2209, S2218
\topsep *498*, E492, F2, F59
\topskip *197*, b354, n61, n129, n184, F1, S128
\totalheight G33, G34, G35
totalpages *703*
\tracefloats S1897
\tracefloatsoff S1897
\tracefloatvals S1897
\traceoff *170*
\traceon *170*
\tracingall *170*, b457
\tracingassigns b486, b519
\tracingcommands b468, b484, b493, b505, b522
\tracingfonts t17, t54, t58, t86, t116, t125, t148, t178, t192, t208, t214, t227, t234, t241, t246, t255, t268, t276, t279
\tracinggroups b477, b530
\tracingifs b478, b529
\tracinglostchars b331, b463, b475, b494, b513, b533
\tracingmacros b467, b483, b495, b512, b532
\tracingnesting b480, b527
\tracingnone b499
\tracingoff t116, t276
\tracingon t117, t277
\tracingonline b452, b504, b521, b542, r627
\tracingoutput ... b453, b508, b525, b539
\tracingpages b462, b474, b494, b514, b534
\tracingparagraphs b464, b476, b495, b511, b531
\tracingrestores b469, b485, b495, b510, b520
\tracingscantokens b459, b479, b502, b528
\tracingstats b461, b473, b493, b515, b535, U2
\triangle x322
\triangleleft x358, x482
\triangleright x359, x482
trivlist (environment) F89
\trivlist *505*, D337, D387, D389, D403, D425, E477, F89, H78, J35, J37
\ttdefault w12, w151, w193, w249, w273, w313, w338, x50

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\ttfamily .. *119*, *378*, w10, w11, w242, w311, w312, w336, w337, z17, D447
ttfamily w252
\ttsubstdefault x20, x32, A32, A110
\twocolumn S199
\twocolumn[] *197*
\typein *72*, *72*
\typein f31
\typeout *72*
\typeout *92*, *702*, a73, a116, a172, a197, a199, a211, a226, a233, a244, a257, a270, a283, a321, c21, c38, c43, c48, f3, f36, f43, f567, f568, f601, f602, f607, j74, n202, n534, n535, n541, n575, n613, n623, n626, r337, w73, w413, w557, w567, w577, x9, x125, M8, M25, P288, P304, P316, P1300, P1499, P1502, R66, R80, S1898, U209, U487, U494, U506, U507, U515

U

\u o237, o386, o469, o583, o590, o610, o617, o748, o1208, o1283, o1284, o1299, o1300, o1309, o1310, o1323, o1324, o1325, o1349, o1350, o1375, o1376, A181, A198
\uccode U173, U181, U188, U190, U193, U195, U438, U446, U453, U455, U458, U460
\Ucharcat w474
\uchyph b332
\ulcdefault s449, s599
\ulcshape *306*, s449, s566, s567, s598, w399, z29
\Umathcode b127, d30, l476, x15, x50, x65, A179, A294, D453, U93, U249, U432
\unboldmath w465
\UndeclareTextCommand *428*, o191, A135, A137, A139, A263, A511, A1039, A1040, A1041, A1042, A1043, A1044
\undefined a9, a11, a17, a57, p92, p93, p94, p95, p96, p97, p98, p99, x116, x117, U101, U113, U114, U134
\undefinedpagestyle O4, O8
\underbar b422, f788, f809
\underbrace x540
\underline *513*
\underline b422, G384, G385
\unexpanded f584, f595, D180, D226, D243, P1328, P1330
\unhcopy b424, H345, I742, I798
\unicodedataline d143, d146, d160, d161, d162
\UnicodeEncodingName o964, o970, o1022, o1033, o1037, o1053, o1055, o1057, o1059, o1060, o1061, o1062, o1063, o1064, o1065, o1066, o1067, o1068, o1069, o1070, o1071, o1072, o1073, o1074, o1075, o1076, o1077, o1078, o1079, o1081, o1082, o1083, o1084, o1085, o1086, o1087, o1088, o1089, o1090, o1091, o1092, o1093, o1094, o1095, o1097, o1098, o1099, o1100, o1101, o1102, o1103, o1104, o1105, o1106, o1107, o1108, o1109, o1110, o1111, o1112, o1113, o1114, o1115, o1116, o1117, o1118, o1119, o1120, o1121, o1122, o1123, o1124, o1125, o1126, o1127, o1128, o1129, o1130, o1131, o1132, o1133, o1134, o1135, o1136, o1137, o1138, o1139, o1140, o1141, o1142, o1143, o1144, o1145, o1146, o1147, o1148, o1149, o1150, o1151, o1152, o1153, o1154, o1155, o1156, o1157, o1158, o1159, o1160, o1161, o1162, o1163, o1164, o1165, o1166, o1167, o1168, o1169, o1170, o1171, o1172, o1173, o1174, o1175, o1176, o1177, o1179, o1180, o1181, o1192, o1193, o1194, o1195, o1196, o1197, o1198, o1199, o1200, o1201, o1202, o1203, o1204, o1205, o1206, o1207, o1208, o1209, o1210, o1211, o1212, o1213, o1214, o1215, o1216, o1217, o1218, A303, A304, A305, A306, A307, A308, A309, A310, A311, A312, A313, A314, A315, A316, A317, A318, A319, A320
\UnicodeFontFile o1020
\UnicodeFontName o1021
\UnicodeFontTeXLigatures ... o976, o1017
\unicoderead d143, d157, d158, d159, d160, d165
uninstall d831
\unitlength *701*, *713*, G53, G64, G73, G83, R252, U82, I5, I29, I30, I32, I34, I42, I43, I44, I45, I60, I63, I73, I74, I84, I85, I93, I94, I107, I108, I119, I164, I176, I241, I256, I316, I318, I332, I340, I342, I357, I375, I377, I392, I397, I399, I414, I417, I422, I480, I481, I508, I509, I537, I538, I612, I628, I648, I654, I690, I691, I693, I694, I697, I698, I700, I701, I712, I713, I715, I716, I718, I719, I721, I722, I750, I751, I753, I754, I757,

I758, I760, I761, I772, I774, I776, I778
\unkern r646
\unless d151, d159, d161
\unlhd w545
\unpenalty . . r649, r653, z116, D419, D441
\unrhd w547
\unsetattribute *39*
\unsetattribute d82, d232
\unvcopy E180
\Uparrow x584
\uparrow x578
\upbracefill x543, x560
\updefault *396*, s556, w21, x94, x101, x103, x111, x113
\Updownarrow x588
\updownarrow x582
\uplus x378
\uppercase U470
\upshape . . . *306*, *307*, *382*, o439, o726, o796, o878, s554, s555, w19, w20, w399, w410, w443, w501, z24, A545
\Upsilon x304
\upsilon x286
use commands:
\use:N g560, g561, g601, Q378, Q382, Q384, Q401
\use:n g142
\use:nn g531, g741
\use_i:nnn *134*
\use_ii:nn *134*
\use_ii_iii:nnn . . Q208, Q221, Q233
\use_iii:nn *134*
\use_iii:nnn Q47
\use_none:n *146*, g7, g394, g410, g529, Q128, Q131, R7
\use_none:nn g699
\usebox G137
\usecounter F225, F238
\usefont *387*, o1530, r80, r282, r623, w522, A7, A48, D457
\UseHook *101*, *103*, *104*, *111*, *113*, *114*, *679*, *681*, *687*, g938, g978, n316, n317, n321, n322, n326, n327, w196, w208, w215, w227, w234, w240, w245, w250, w512, D162, D174, D177, D201, D204, D208, D211, P743, P744, P747, P748, P770, P771, P774, P775, Q166, Q167, Q169, Q170, R82, R104, R297
\UseLegacyTextSymbols A502, A709
\UseOneTimeHook *101*, *103*, *113*, g938, g979, n17, n59, n72, D13, D17, D27, D28, D30
\usepackage *104*, *119*, *679*, *680*, *681*, *703*, P499, P545, P1315
\UseRawInputEncoding . U279, U335, U382
\UseTextAccent . *223*, o149, o150, o188, A131, A132, A133, A176, A178, A848, A1033, A1034, A1036, A1037
\UseTextSymbol *223*, o150, o186, A130, A286, A847, A916

V

\v o238, o387, o468, o586, o587, o588, o592, o594, o597, o599, o601, o607, o613, o614, o615, o619, o621, o624, o626, o628, o634, o749, o1213, o1293, o1294, o1295, o1296, o1305, o1306, o1339, o1340, o1345, o1346, o1357, o1358, o1365, o1366, o1369, o1370, o1392, o1393, o1394, o1395, o1396, o1397, o1398, o1399, o1400, o1401, o1402, o1405, o1406, o1407, o1408, o1411, o1412, A182, A199
\vadjust l37, l59, l71, l100, l107, l341, l357, l375, l391, L201, L223
\valign A122, A839
\value *258*
\value p14, N9
\varbigtriangledown x362, x365
\varbigtriangleup x363, x364
\varepsilon m15, x291
\varphi x296
\varpi x293
\varrho x294
\varsigma x295
\vartheta x292
\vbadness *710*, b317, R136, R138, R187, R189, S2206
\vbox *700*, *710*
vbox commands:
\vbox_set_to_ht:Nnn R139, R190
\vbox_to_zero:n R250
\vbox_unpack:N R150, R200
\vdash x404
\vdots x510
\vec x524
\vector j219, I233, I817, I834
\vee x367, x369
\verb D451, D476, D489, D495, D504, D510, D520, D525, D538, D540
verbatim (environment) D445
\verbatim D445
verbatim* (environment) D469
\verbvisiblespace D451, D453, D460, D464, D476, D481
\Vert x571, x573

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx

\vert x576
\vfil b411, A123, A126, A840, A843, R308, R320, S175, S194, S412, S459, S627, S686, I566, I578
\vfilneg b411
\vfuzz *710*, b340, O61, O68, R135, R137, R186, R188
\vglue b401
\vline H366
\vphantom o494, o510, E75
\vrule *790*, b405, l452, o293, o295, o499, o515, t144, x558, x559, x561, x562, G146, G148, G199, G206, G383, G427, H186, H219, H347, H366, S1851, S2226, S2259, I227, I299, I302, I324, I333, I350, I359, I383, I391, I406, I413, I565, I578, I726, I782
\vskip *178*, *185*, *185*
\vspace l330, l400, l401, l402
\vsplit S382, S429, S2208

W

\wedge x366, x368
\whatsit *38*, d186
\widehat x527
\widetilde x526
\widowpenalties b106
\widowpenalty b324, r636
\width G30
\wlog *645*, a100, b40, b145, b239, b254, b284, b299, d6, d7, d8, d54, P318, U46, U540
\wp x312
\wr x382
\write *701*, *708*

X

\x r304, r305
\XeTeXcharclass r611, U25, U33, U40, U53, U59, U68, U75
\XeTeXcharclassCL U106
\XeTeXcharclassCM U110
\XeTeXcharclassEX U107
\XeTeXcharclassID U104
\XeTeXcharclassIS U108
\XeTeXcharclassNS U109
\XeTeXcharclassOP U105
\XeTeXcharglyph o1018
\XeTeXdashbreakstate U206
\XeTeXglyph o1018
\XeTeXintercharclasses U100, U133
\XeTeXinterchartoks U101, U115, U116, U117, U118, U119, U120, U121, U122, U123, U124, U125, U126, U127, U128, U129, U134, U139, U140, U141, U142, U143, U144, U145, U146, U147, U148, U149, U150, U151, U152, U153
\XeTeXmathcode U94, U433
\XeTeXrevision U27
\XeTeXuseglyphmetrics U203, U205
\XeTeXversion w473, U27
\Xi x301
\xi x281
\xtxHanGlue U113, U137, U145, U146, U147, U148, U149, U150, U151, U152, U153
\xtxHanSpace U114, U138, U139, U140, U141, U142, U143, U144

Y

\year a185, c14, P1011, P1142, P1231

Z

\Z U185, U429, U450
\z U176, U430, U441
\zeta x273

File Key: a=ltdirchk.dtx, b=ltplain.dtx, c=ltvers.dtx, d=ltluatex.dtx, e=ltexpl.dtx, f=ltdefns.dtx, g=lthooks.dtx, h=ltalloc.dtx, i=ltcntrl.dtx, j=lterror.dtx, k=ltpar.dtx, l=ltspace.dtx, m=ltlogos.dtx, n=ltfiles.dtx, o=ltoutenc.dtx, p=ltcounts.dtx, q=ltlength.dtx, r=ltfssbas.dtx, s=ltfssaxes.dtx, t=ltfsstrc.dtx, u=ltfsscmp.dtx, v=ltfssdcl.dtx, w=ltfssini.dtx, x=fontdef.dtx, y=preload.dtx, z=ltfntcmd.dtx, A=lttextcomp.dtx, B=ltpageno.dtx, C=ltxref.dtx, D=ltmiscen.dtx, E=ltmath.dtx, F=ltlists.dtx, G=ltboxes.dtx, H=lttab.dtx, I=ltpictur.dtx, J=ltthm.dtx, K=ltsect.dtx, L=ltfloat.dtx, M=ltidxglo.dtx, N=ltbibl.dtx, O=ltpage.dtx, P=ltclass.dtx, Q=ltfilehook.dtx, R=ltshipout.dtx, S=ltoutput.dtx, T=lthyphen.dtx, U=ltfinal.dtx